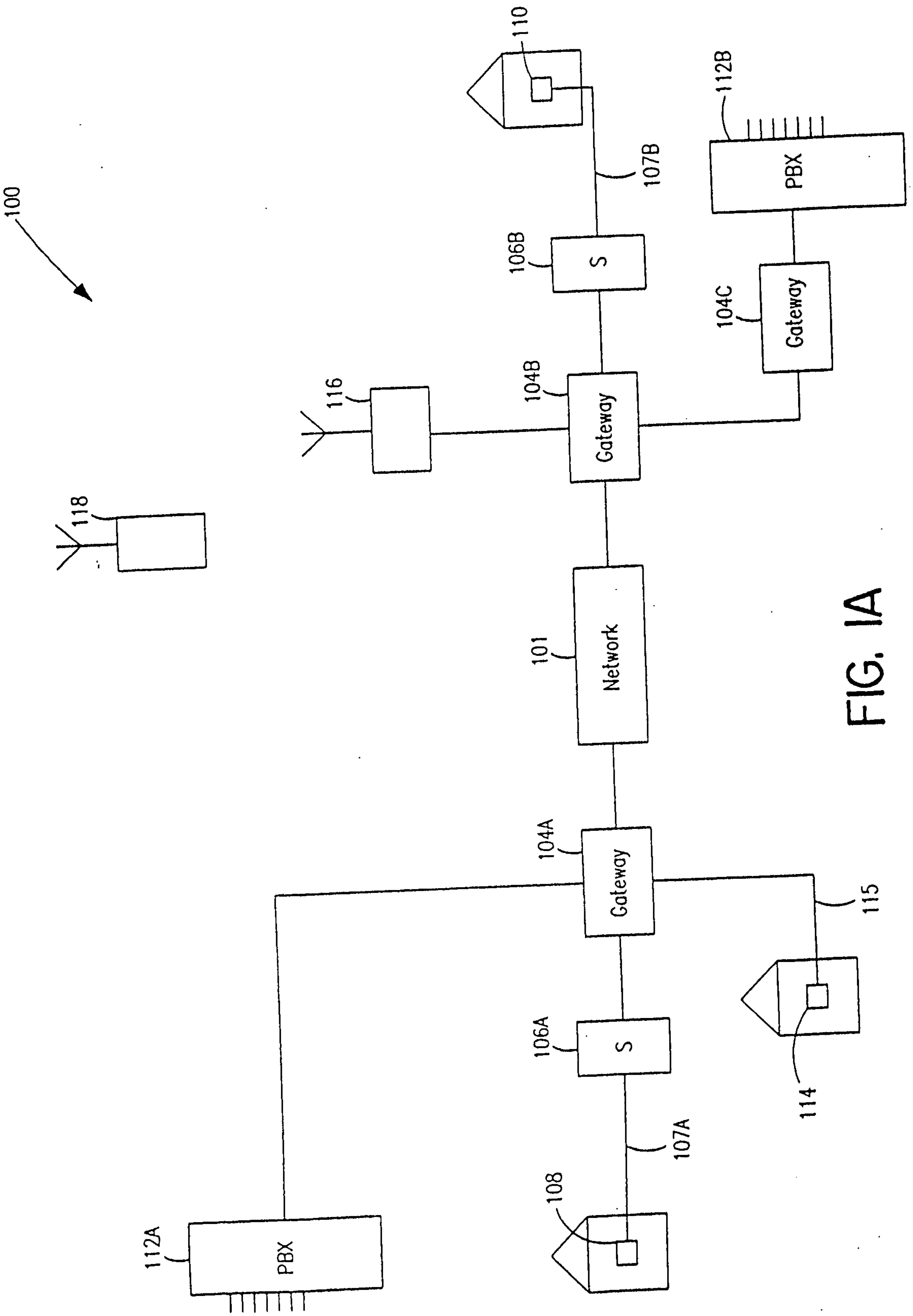

100
110
112B
PBX
107B
106B
S
104C
Gateway
104B
Gateway
116
118
101
Network
104A
Gateway
115
114
106A
S
107A
108
112A
PBX

FIG. IA

Blakely, Sokoloff, Taylor & Zafman LLP (714) 557-3800
Title: Methods for Power Reduction in a Digital Signal Processor Integrated Circuit
1st Named Inventor: Ruban Kanapathippillai
Application No.: 10/645,366 Docket No.: 42P14037D
Sheet: 2 of 64

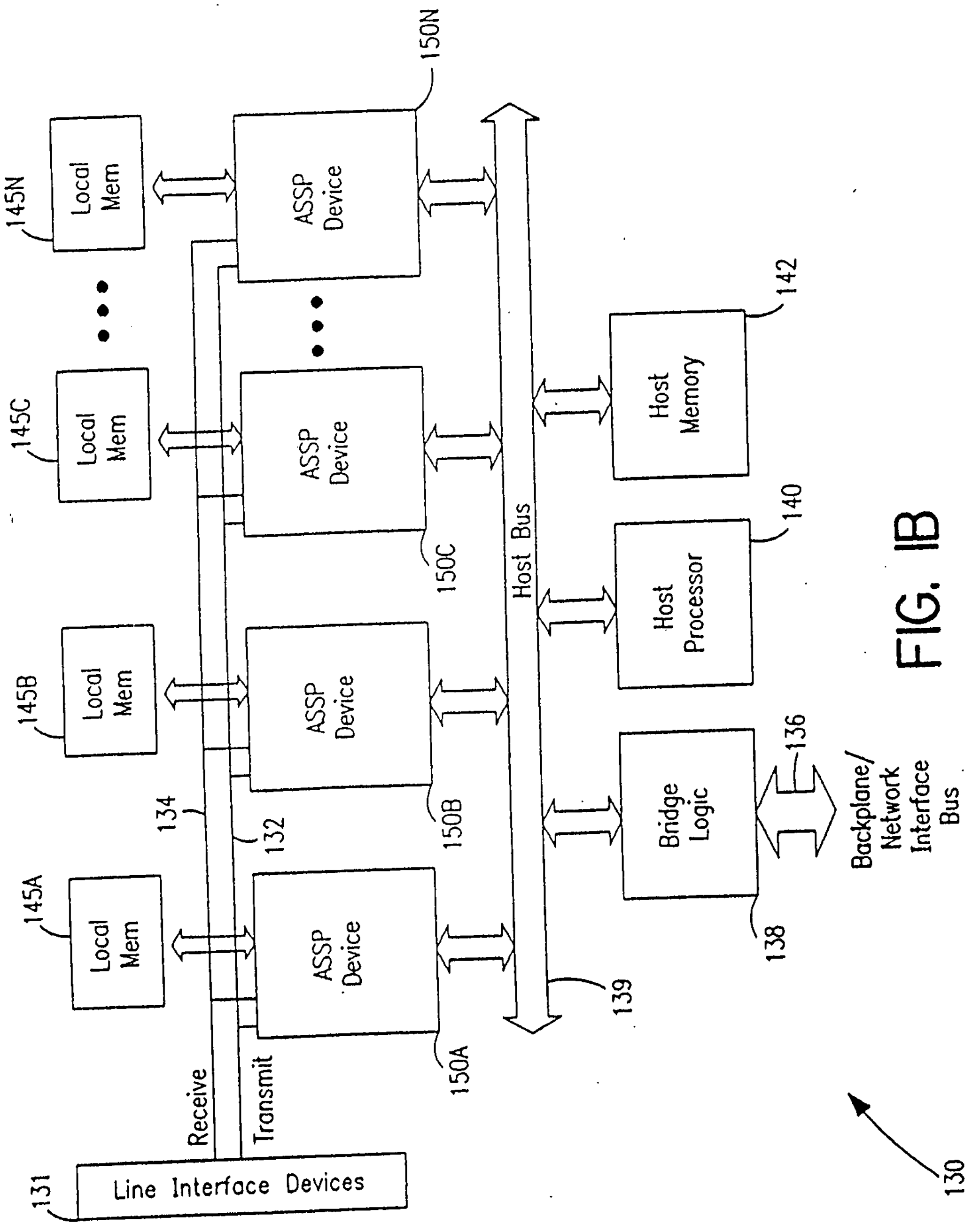


FIG. IB

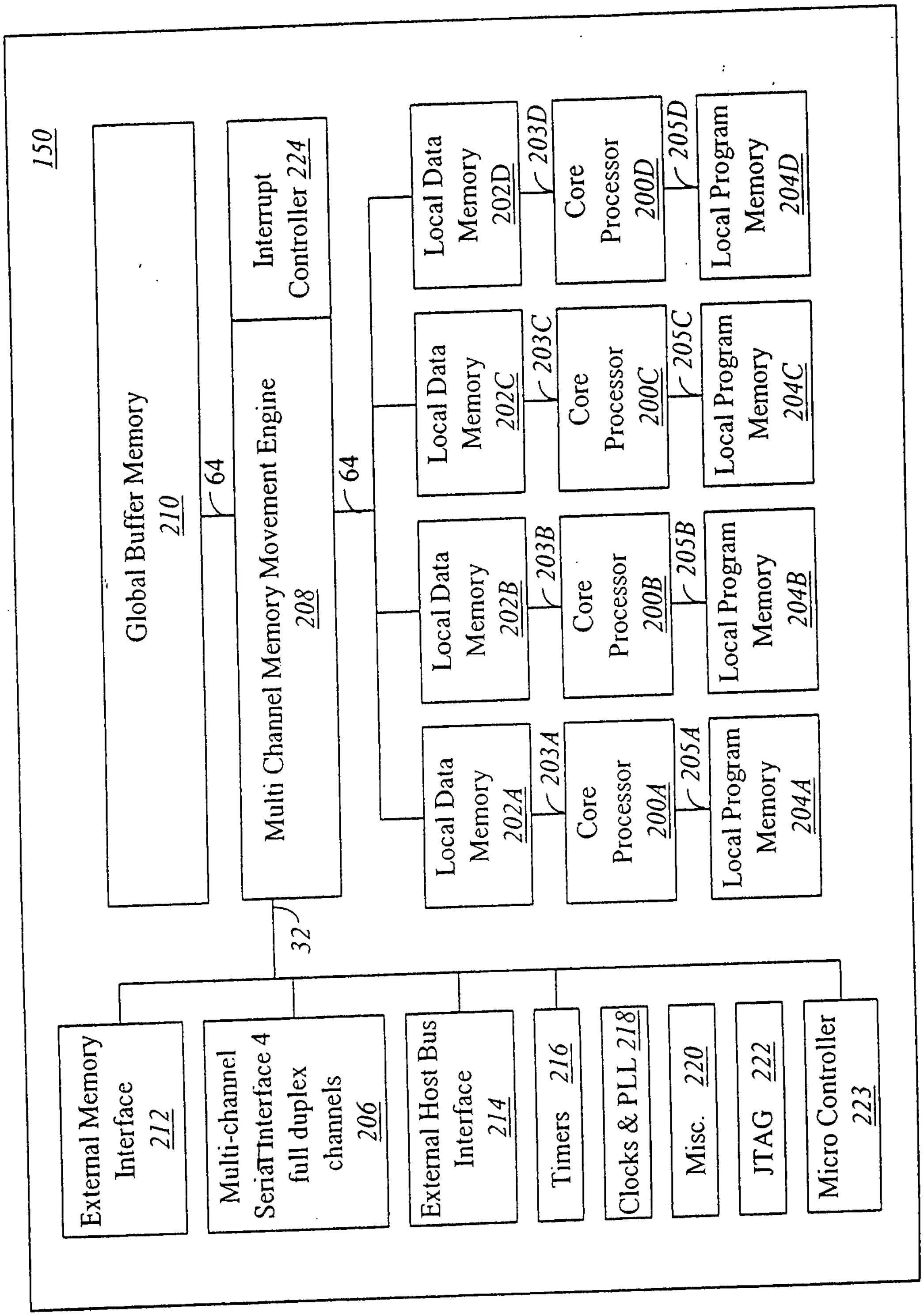

150
Global Buffer Memory 210
64
Interrupt Controller 224
Multi Channel Memory Movement Engine 208
64
Local Data Memory 202D
203D
Core Processor 200D
205D
Local Program Memory 204D
Local Data Memory 202C
203C
Core Processor 200C
205C
Local Program Memory 204C
Local Data Memory 202B
203B
Core Processor 200B
205B
Local Program Memory 204B
Local Data Memory 202A
203A
Core Processor 200A
205A
Local Program Memory 204A
32
External Memory Interface 212
Multi-channel Serial Interface 4 full duplex channels 206
External Host Bus Interface 214
Timers 216
Clocks & PLL 218
Misc. 220
JTAG 222
Micro Controller 223

FIG. 2

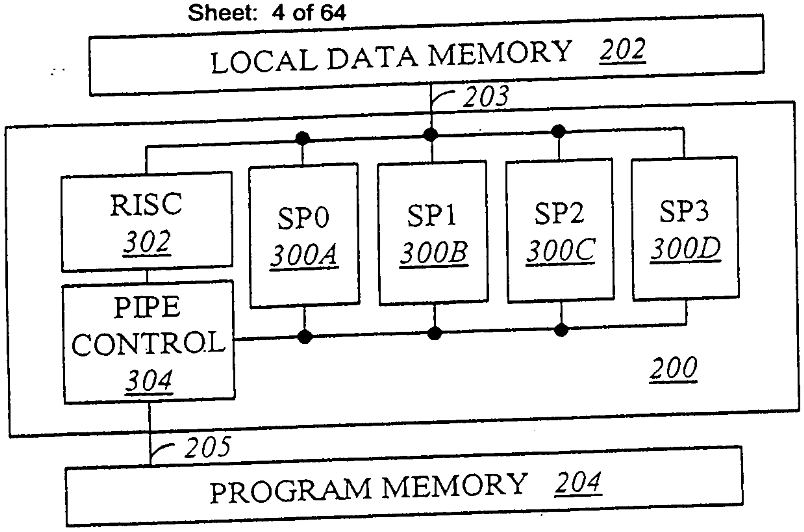


FIG. 3

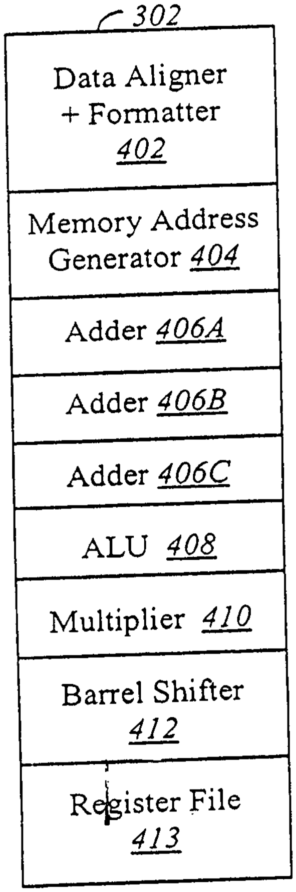


FIG. 4

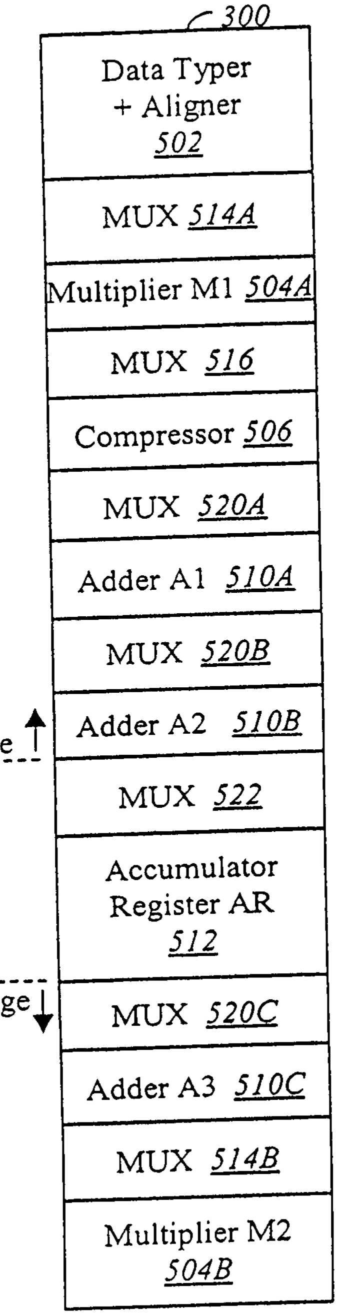


FIG. 5A

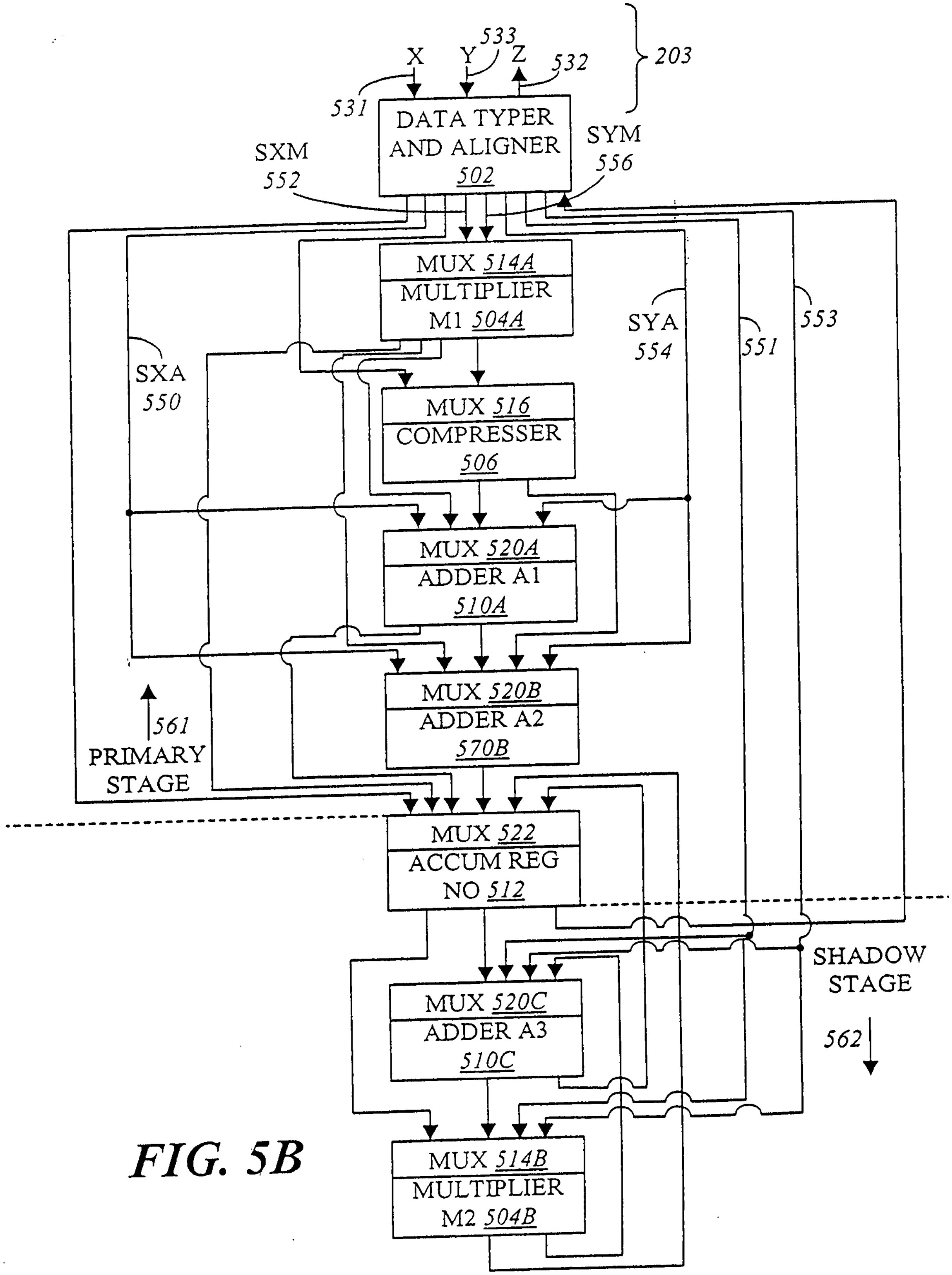

X
Y
Z
533
532
531
203
DATA TYPER AND ALIGNER 502
SXM 552
SYM 556
MUX 514A
MULTIPLIER M1 504A
SYA 554
551
553
SXA 550
MUX 516
COMPRESSER 506
MUX 520A
ADDER A1 510A
MUX 520B
ADDER A2 570B
561
PRIMARY STAGE
MUX 522
ACCUM REG NO 512
SHADOW STAGE
562
MUX 520C
ADDER A3 510C
MUX 514B
MULTIPLIER M2 504B

FIG. 5B

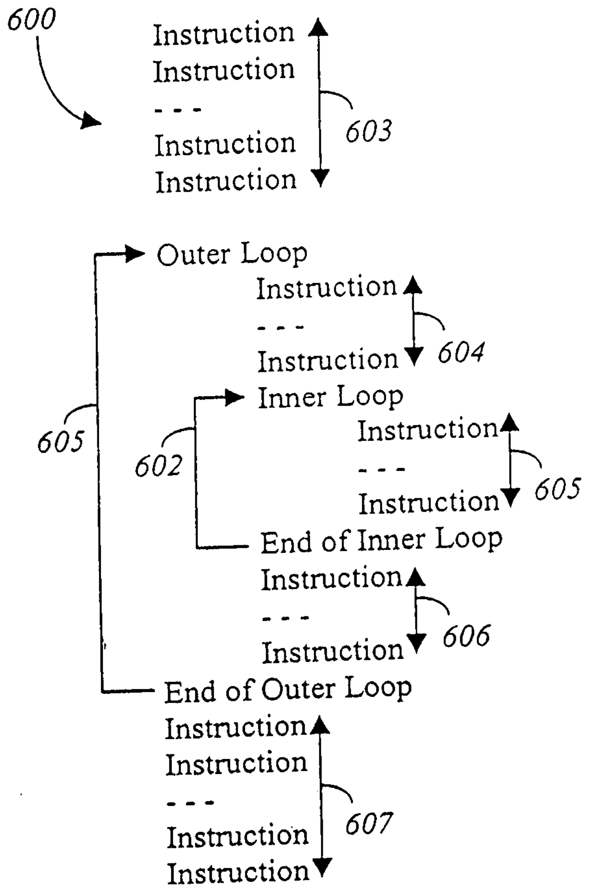
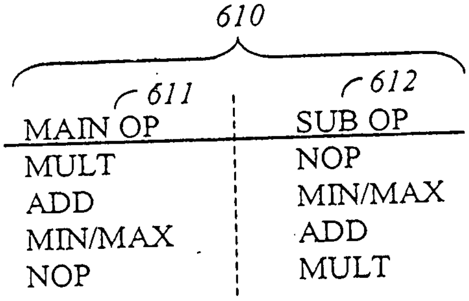


FIG. 6A



MAIN OP	SUB OP
MULT	NOP
ADD	MIN/MAX
MIN/MAX	ADD
NOP	MULT

FIG. 6G

20-bit ISA

39	19		
0	0	20-bit parallel	Control II Control
0	1	20-bit serial	Control # Control
1	0	40-bit extended	DSP, extensions/Shadow
1	1	20-bit serial	DSP # DSP

FIG. 6B

6-bit operand specifier:
A 6-bit specifier is used in DSP extended instructions to access memory and register operands.

5	4	3	2	1	0

M/R

0	0	ac-page	ereg
0	1	gpr:0-r15	GPR
1	ptr(r0)to(r15)	off	Mem[ptr{0-15}]II PTR[0-15]+=offset1/offset2 Always postupdate

This allows access to data memory, ereg and GPR

- Bit 5=1:Use rX(X:0-7) register to obtain effective memory address and post-modify the
  ptr field by one of two possible offsets specified in rX registers.
  dmem[ptr], ptr=ptr+offset1, if off=0
  ptr=ptr+offset2, if off=1

- Bit 5=0:Access ac-page or GPR

If Bit-4 is set to 0, then bits 3:0 control access to the general-purpose register file (r0-15) or to execution unit registers.

GPR	GPR Intr page	ac-page	ac intr page	ereg-Shadow DSP
R0	R0	A0	A0_I	A0
R1	R1	A1	A1_I	A1
R2	R2	T	T	T
R3	R3	TR	TR	TR
R4	R4			
R5	R5			
R6	R6			
R7	R7			
R8	R8			SX1
R9	R9			SX1s
R10	R10			SX2
R11	R11			SX2s
R12	R12_i			SY1
R13	R13_i			SY1s
R14	R14_i			SY2
R15	R15_i			SY2s

FIG. 6C

4-bit operand specifier:
Memory operands: (rX) specifies an access out of the data memory to the extension unit for the function that needs to be performed. The address for the access is specified in the rX register in the general register file that hold the 14-bit pointer (16K of addressing) to memory, 5-bit signed offset or a 3-bit unsigned offset that can post-modify the address. In addition, each pointer is typed for efficient SIMD processing and includes a permute control for rearranging data elements of a vector on the fly. The "podi" core can deal with 4-element 16-bit real vectors or complex data directly. This ability to manipulate memory data directly reduces the instruction width greatly and allows efficient signal processing.

(rX): Memory Address Registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
type				cb	x	permute				off1:(0-7			off0:(-16 to 15)					ptr: pointer													

Fig. 6D

For shadow DSP instructions, the 3-bit specifier for operands is defined as follows:

2	1	0	
0	0	0	A0
0	0	1	A1
0	1	0	T
0	1	1	TR
1	0	0	SX1
1	0	1	SX1s
1	1	0	SX2
1	1	1	SX2s

EREG1

2	1	0	
0	0	0	A0
0	0	1	A1
0	1	0	T
0	1	1	TR
1	0	0	SY1
1	0	1	SY1s
1	1	0	SY2
1	1	1	SY2s

EREG2

Only the shadow DSP instructions can see the above modified page of execution unit registers.

FIG. 6E

5–bit operand specifier:

The 5–bit specifier includes the 4–bit specifier for general data operands and the special purpose registers. It is used in RISC instructions.

4	3	2	1	0
0	spr:s0–s15			
1	gpr:r0–r15			

	SPR	Intr page	
	SPR	SPR intr page	
0	fu–ctl	fu–ctl_I	
1	a–type	a–type_I	
2	ps–ctl	ps–ctl	
3	t–type	t–type	
4	pl–ctl	pl–ctl	
5	cb–ctl	cb–ctl_I	
6	shuffle	shuffle	
7	lo–ptr	lo–ptr	
8	status	status_I	
9	loop–ctl	loop–ctl	
10	pcr	pcr	stack(8)
11	reserved	reserved	
12	reserved	reserved	
13	reserved	reserved	
14	reserved	reserved	

NOTE: All SPR registers are reset to all zeros at power on reset except for the PCR register.

FIG. 6F

DSP instructions

	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	
Multiply	1	0	0	PS		S'	SX				SY				V/S	SA	DA	Sub-op			
							da=sx*sy											0	0	0	Nop
							da=(sx*sy)+sa											0	0	1	Add
							da=(sx*sa)+sy											0	1	0	Add
							da=(sx*sy)+sa											0	1	1	Sub
							da=(sx*sa)+sy											1	0	0	Sub
							da=min(sx*sy,sa)											1	0	1	Min
							da=min(sx*sa,sy)											1	1	0	Min
							da=max(sx*sy,sa)											1	1	1	Max
Add	1	0	1	PS		+/–	SX				SY				V/S	SA	DA	Sub-op			
							da=sx+sy											0	0	0	Nop
							da=sx+sy+sa											0	0	1	Add
							da=sx+sy: sa=sx+sy:											0	1	0	AddSub
							da=(sx+sy)*sa											0	1	1	Mul
							da=–(sx+sy)*sa											1	0	0	MulN
							da=min(sx+sy,sa)											1	0	1	Min
							da=max(sx+sy,sa)											1	1	0	Max
							da=ssum(sa) (sx,sy unused)											1	1	1	CombAdd
Extremum	1	1	0	PS		x/n	SX				SY				V/S	SA	DA	Sub-op			
							da=ext(sx,sy)											0	0	0	Nop
							da=ext(sx,sy,sa)											0	0	1	Ext
							da=ext(sx,sa)*sy											0	1	0	Mul
							da=–ext(sx,sa)*sy											0	1	1	MulN
							da=ext(sx,sa)+sy											1	0	0	Add
							da=ext(sx,sa)–sy											1	0	1	Sub
							ext(sa,da) ?1=sx,tr=sy,lcs=lc											1	1	0	amax
type–match	1	1	0	PS		0	SX				SY				x	x	x	1	1	1	
	1	1	0	PS		0	x	x	x	x	x	x	x	x	x	x	x	1	1	1	
Permute	1	1	0	PS		1	Type				SY				0	ereg		1	1	1	Permute
Reserved	1	1	1	PS		x	SX				SY				SA	DA	V/S	Sub-op			

FIG. 6H

Control and specifier Extensions

	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Mul	0	Pred		PL		Sxt		Syt		Rnd	Lt	S'	S'	S'	0	SA	DA	abs	0	0	
Add	0	Pred		PL		Sxt		Syt		Lt	Sub-ext				0	SA	DA	abs	0	0	
											+/-	+/-	+/-	x							Nop (uadd)
											x	V/S	Rnd	Fp							Mul/MulN
											tr/ctl		Gx	Fp							Min/Max
Ext	0	Pred		PL		Sxt		Syt		tr-ctl		Gx	Sub-ext		0	SA	DA	abs	0	0	
													Lt	Fp							Add/sub
													Rnd	V/S							Mul
	0	Pred		PL		Sxt		Syt		Pctl1					0	ereg		Pcll	0	0	

Type/offset/permute extensions

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	Pred		PL		x	Type:SX				Type:SY				0	SA	DA	x	0	1	Type override
0	Pred		PL		Psx	Permute:SX				Permute:SY				0	SA	DA	Psy	1	0	permute override
0	Pred		I/R	I/R	prX	Offset:SX				Offset:SY				0	SA	DA	prY	1	1	Offset override

Shadow DSP

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Op		PL		op	SA	ereg1			DA	ereg2			1	SA	DA	Sub-op		

nop

1	1	0	PL		0	x	x	x	Rnd	x	x	x	x	0	SA	DA	1	1	1

FIG. 6I

Control instructions

	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
add,sub	L	Pred		0	0	0	RX				RY				RZ				+/-	0
max,min	L	Pred		0	0	0	RX				RY				RZ				X/N	1
Shift	L	Pred		0	0	1	RX				UI4				RZ				UI2	R/L
Logic	L	Pred		0	1	0	RX				RY				RZ				&,	&1
Mux	L	Pred		0	1	1	RX				RY				RZ				Pd	0
mov	L	Pred		0	1	1	RX				DZ				Rxt	DzI	0	0	0	1
addi	L	Pred		0	1	1	SI4				DZ				x	x	1	0	0	1
mov2erg	L	Pred		0	1	1	RX				unit		ereq		qd	type		1	0	1
l.dm	L	Pred		0	1	1	RX				DZ1				DZ2				1	1
Set4bits	L	Pred		1	0	0	UI4:POS				RZ				Rzt	UI4				0
Set2bits	L	Pred		1	0	0	UI4:POS				RZ				Rzt	UI2		0	0	1
Setbit	L	Pred		1	0	0	UI4:POS				RZ				Rzt	UI1	UI1	1	0	1
Movi	L	Pred		1	0	0	SI8								RZ				1	1
Jmp	L	Pred		1	0	1	SI9									0	PRED		0	0
Call	L	Pred		1	0	1	SI9									1	PRED		0	0
Loop	L	Pred		1	0	1	UI5:Lcount					UI5:Lsize					UI2:Lst		0	1
Jmpi	L	Pred		1	0	1	RX				x	x	x	x	x	0	PRED		1	0
Calli	L	Pred		1	0	1	RX				x	x	x	x	x	1	PRED		1	0
Loopi	L	Pred		1	0	1	RX				x	UI5:Lsize					UI2:Lst		1	1
Test	L	Pred		1	1	0	RX				RY				PZ			=,<,>		0
Testbit	L	Pred		1	1	0	RX				UI5					PZ		B	0	1
Andp, orp	L	Pred		1	1	0	Pa			Pb			Pc		PZ			&	1	1
Load	L	Pred		1	1	1	MX				RZ				Ext			0	0	0
Store	L	Pred		1	1	1	MZ				RZ				Ext			1	0	0
eLoad	L	Pred		1	1	1	MX				RY				1	1	1	0	0	0
eStore	L	Pred		1	1	1	MZ				RY				1	1	1	1	0	0
Extended	L	Pred		1	1	1	Bits 27:16												1	0
Logic2	L	Pred		1	1	1	RX				RY/RZ				Rxt	Ryt	&,1,&1,I		0	1
mov-erg	L	Pred		1	1	1	unit		ereq		RZ				qd	Sft		0	1	1
Crb	L	Pred		1	1	1	RX				RZ				s/m	0	0	1	1	1
Panty	L	Pred		1	1	1	RX				PZ			O/E	0	1	0	1	1	1
Stm	L	Pred		1	1	1	MZ				RX				1	1	0	1	1	1
Abs	L	Pred		1	1	1	RX				RZ				0	0	1	1	1	1
Neg	L	Pred		1	1	1	RX				RZ				0	1	1	1	1	1
Div-step	L	Pred		1	1	1	RX				RZ				1	0	1	1	1	1
Test&Set	L	Pred		1	1	1	RX				PZ			0	1	1	1	1	1	1
Reserved	L	Pred		1	1	1						0	0	1	1	1	1	1	1	1
Return	L	Pred		1	1	1	Pred		I-ctl		0	1	0	1	1	1	1	1	1	1
Zero-ac	L	Pred		1	1	1	ac #				1	1	0	1	1	1	1	1	1	1
eSync	L	Pred		1	1	1	RZ				0	1	1	1	1	1	1	1	1	1
Swi	L	Pred		1	1	1	UI3			0	1	1	1	1	1	1	1	1	1	1
Nop	L	Pred		1	1	1	UI3			1	1	1	1	1	1	1	1	1	1	1

<Bit1,Bits9-6> ==UI5 (Shift Amount)

<Bit3,Bits13-10>==UI5 POS

FIG. 6J

Extended Control

	Bits 13:2 of upper half (39:20)																																
	13	12	11	10	9	8	7	6	5	4	3	2	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Insert/Extract	RX				RZ				0	0	0	0	0	x	x	0	Rxt	Rzt	I/E	R/I	R/I	Offset:UI5					Length:UI5					0	
																						x	RY				RV				x		
Inserti	UI4 length				RZ				0	0	0	1	0	x	x	0	Rzt	UI5 Position					Imm10										
Shift	RX				RZ				0	0	0	0	0	rxh	rxl	0	rzh	rzt	D	U/S	1	Shift UI5					A/L	Lt	R/L	0	Fill	1	Fill: Sign/
																					0	ryh	RY										Zero
Rotate	RX				RZ				0	0	0	0	0	x	x	0	x	x	x	x	1	Shift UI5					1	x	R/L	1	x	1	
																					0	ryh	RY										
jmp, call	UI7							J/C	0	0	1	0	0	Pred		0	x	UI15															
dloop	UI4 outer LC				UI4 inner LC				0	0	1	1	0	x	exit	0	UI1	UI4 outer Lsize				UI4 inner Lsize				UI2 C-Ls		UI4 inner Lstart				0	Bit 15 is
dloopi	RX				RY				0	0	1	1	0	x	exit	0	x	UI4 outer Lsize				UI4 inner Lsize				UI2 C-Ls		UI4 inner Lstart				1	continuation
mult	RX				RY				0	1	0	0	0	x	x	0	0	rxh	md	ryh	+/-	=/-	RZ				i/f	rzh	rzl	s/u	s/u	0	of inner LC
add/sub	RX				RY				0	1	0	0	0	x	x	0	0	rxh	rxl	ryh	ryl	+/-	RZ				Lt	rzh	rzl	x	x	1	
Reserved									0	1	0	0	0	x	x	0	1	x	x	x	x											0	
logicp	RX				D	PZ			0	1	0	0	0	x	x	0	1	T/F	T/F	T/F	&i	&i	PY				PV				x	1	andp,orp,
Testi	RX				D	PZ			0	1	0	1	0	=>,<		0	Imm16																andorp,
Movi	H/L	Fill		RZ					0	1	1	0	0	x	x	0	Imm16																orandp,
loadi	Type			RZ					0	1	1	1	0	x	x	0	0	0	Imm14														pz=(px
storei	Type			RX					0	1	1	0	0	x	x	0	0	1	Imm14														relop py)
loadt	RX				RZ				0	1	1	1	0	x	x	0	1	Rzt	0	Type			SI10										relop pv)
storet	M7				RX				0	1	1	1	0	x	x	0	1	Rzt	1	Type			~~SI10~~										
Addi/subi	RX				RZ				1	0	+/-	0	0	L1	s/u	0	Imm16																
mini,maxi	RX				RZ				1	0	X/N	1	0	x	x	0	Imm16																
andi,on	RX				RZ				1	1	&i	H/L	0	x	x	0	Imm16																

FIG. 6K

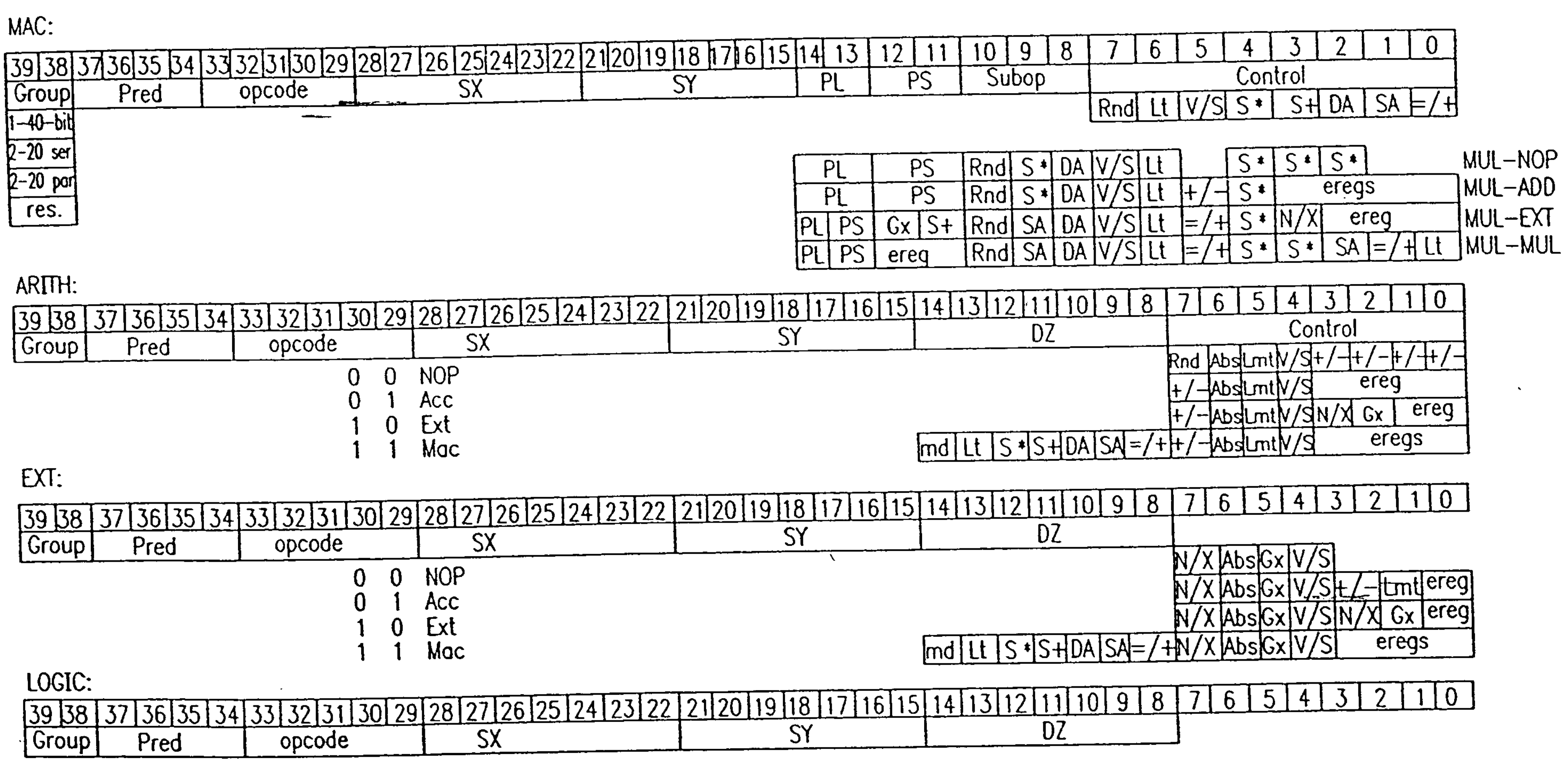

MAC:
39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Group
Pred
opcode
SX
SY
PL
PS
Subop
Control
Rnd Lt V/S S* S+ DA SA =/+
1-40-bit
2-20 ser
2-20 par
res.
PL PS Rnd S* DA V/S Lt S* S* S*
MUL-NOP
PL PS Rnd S* DA V/S Lt +/– S* eregs
MUL-ADD
PL PS Gx S+ Rnd SA DA V/S Lt =/+ S* N/X ereg
MUL-EXT
PL PS ereg Rnd SA DA V/S Lt =/+ S* S* SA =/+ Lt
MUL-MUL
ARITH:
39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Group
Pred
opcode
SX
SY
DZ
Control
Rnd Abs Lmt V/S +/– +/– +/– +/–
+/– Abs Lmt V/S ereg
+/– Abs Lmt V/S N/X Gx ereg
md Lt S* S+ DA SA =/+ +/– Abs Lmt V/S eregs
0 0 NOP
0 1 Acc
1 0 Ext
1 1 Mac
EXT:
39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Group
Pred
opcode
SX
SY
DZ
N/X Abs Gx V/S
N/X Abs Gx V/S +/– Lmt ereg
N/X Abs Gx V/S N/X Gx ereg
md Lt S* S+ DA SA =/+ N/X Abs Gx V/S eregs
0 0 NOP
0 1 Acc
1 0 Ext
1 1 Mac
LOGIC:
39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Group
Pred
opcode
SX
SY
DZ

FIG. 6L-1

Shift:

39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Group		Pred				opcode				SX								DZ							Amount						PL		PS	Lt	Rot	Fill	A/L	1	1	Shift
Group		Pred				opcode				SX								DZ							Amount						Position						I/E	0	1	Insert/extract
Group		Pred		Imm2		opcode				DZ				Imm14															Length				Position						0	Setbits

Immediate:

39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Group		op	DZ					Imm32																															
Group		opcode			SX							DZ						Subop MOV ADD SUB MIN MAX AND OR			Imm16																		

Test:

39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Group		Pred				opcode					SX							SY							DPz				Subop										

Branch:

39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Group		Pred				opcode														Imm20																			

Misc:

FIG. 6L-2

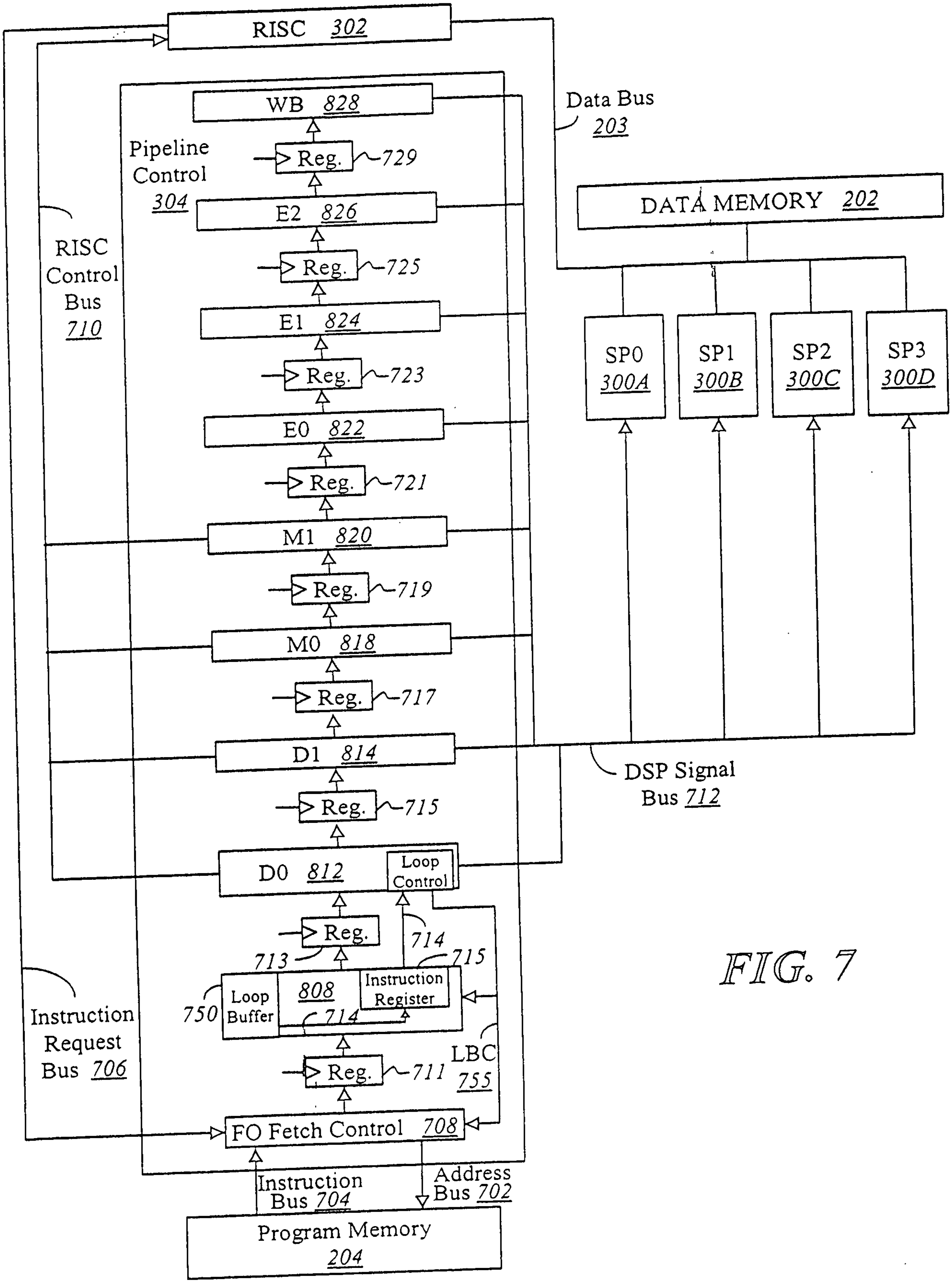

RISC 302
WB 828
Pipeline Control 304
Reg. 729
E2 826
Reg. 725
E1 824
Reg. 723
E0 822
Reg. 721
M1 820
Reg. 719
M0 818
Reg. 717
D1 814
Reg. 715
D0 812
Loop Control
Reg.
713
714
715
Loop Buffer
808
Instruction Register
714
750
LBC 755
Reg. 711
FO Fetch Control 708
Instruction Bus 704
Address Bus 702
Program Memory 204
RISC Control Bus 710
Instruction Request Bus 706
Data Bus 203
DATA MEMORY 202
SP0 300A
SP1 300B
SP2 300C
SP3 300D
DSP Signal Bus 712

FIG. 7

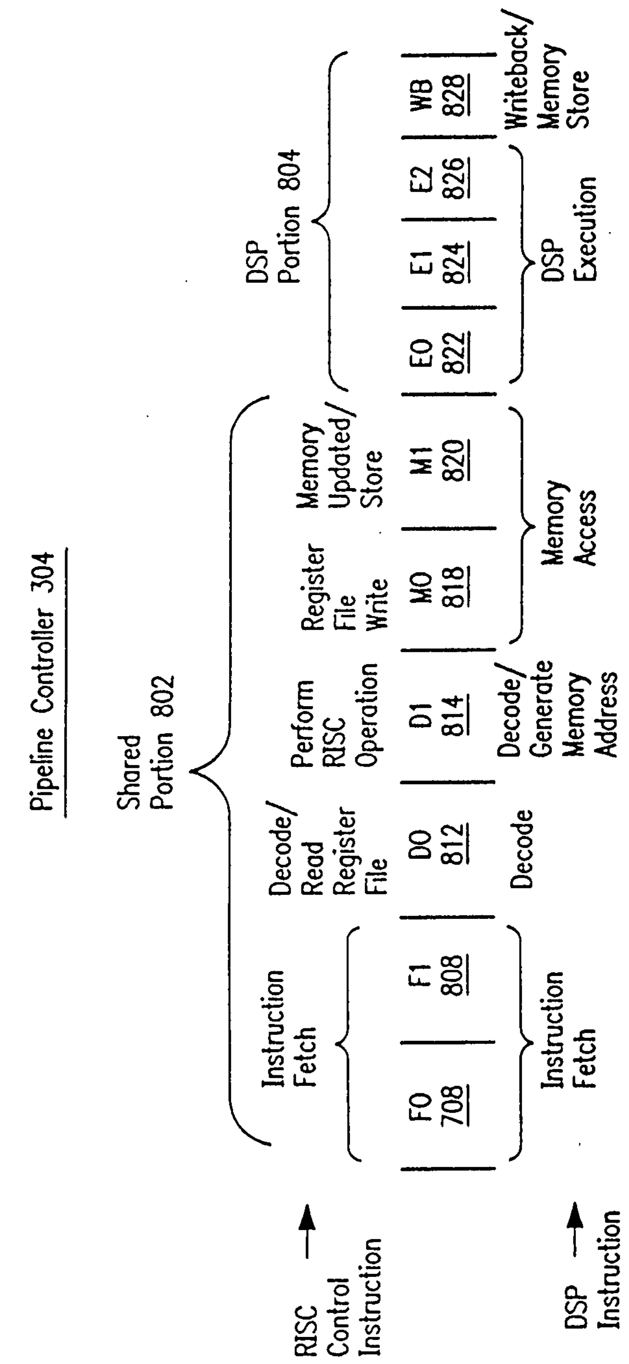

Pipeline Controller 304
Shared Portion 802
DSP Portion 804
Instruction Fetch
Decode/ Read Register File
Perform RISC Operation
Register File Write
Memory Updated/ Store
F0 708
F1 808
D0 812
D1 814
M0 818
M1 820
E0 822
E1 824
E2 826
WB 828
Writeback/ Memory Store
Instruction Fetch
Decode
Decode/ Generate Memory Address
Memory Access
DSP Execution
RISC Control Instruction
DSP Instruction

FIG. 8a

Blakely, Sokoloff, Taylor & Zafman LLP (714) 557-3800
Title: Methods for Power Reduction in a Digital Signal Processor Integrated Circuit
1st Named Inventor: Ruban Kanapathippillai
Application No.: 10/645,366 Docket No.: 42P14037D
Sheet: 18 of 64

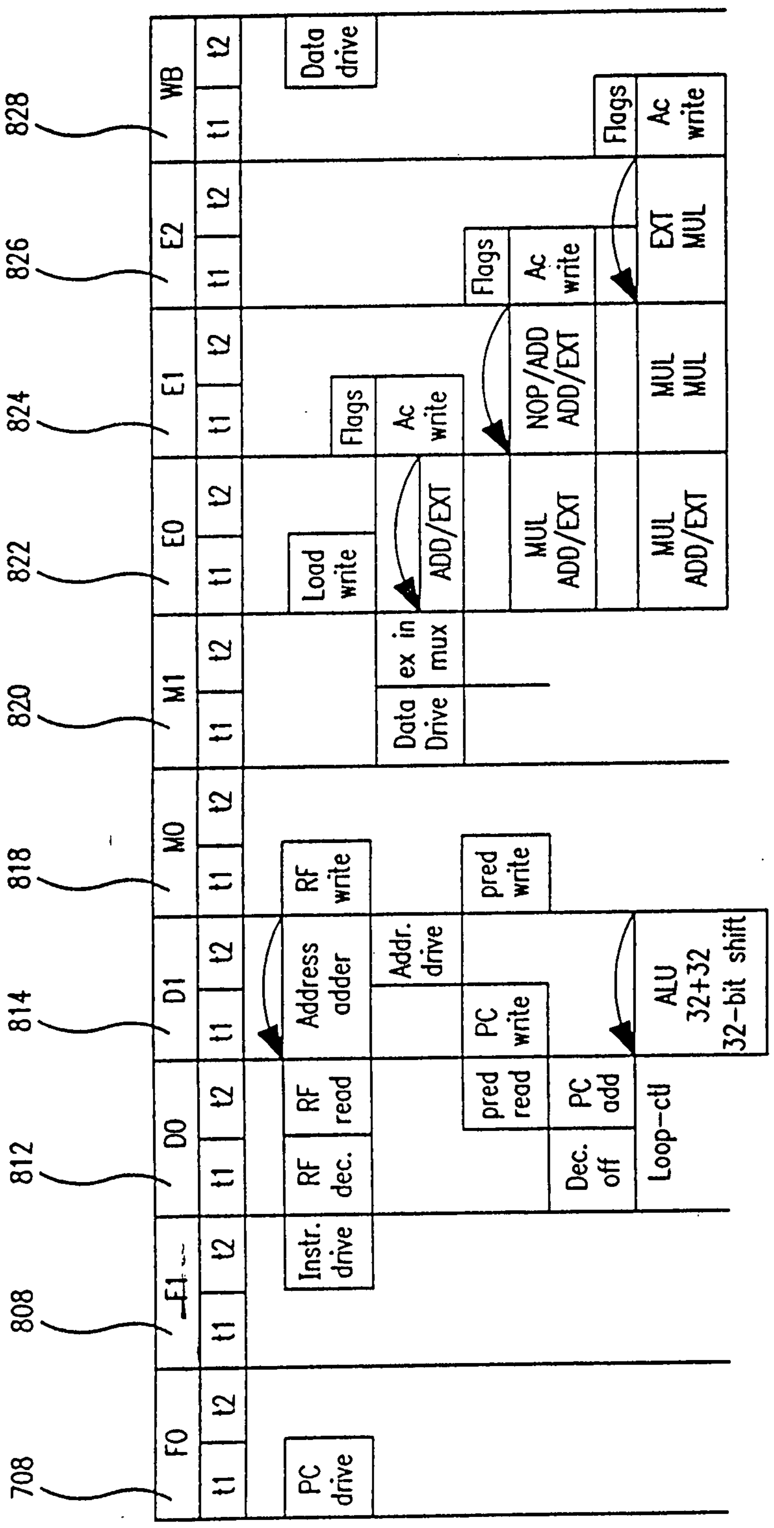


FIG. 8b

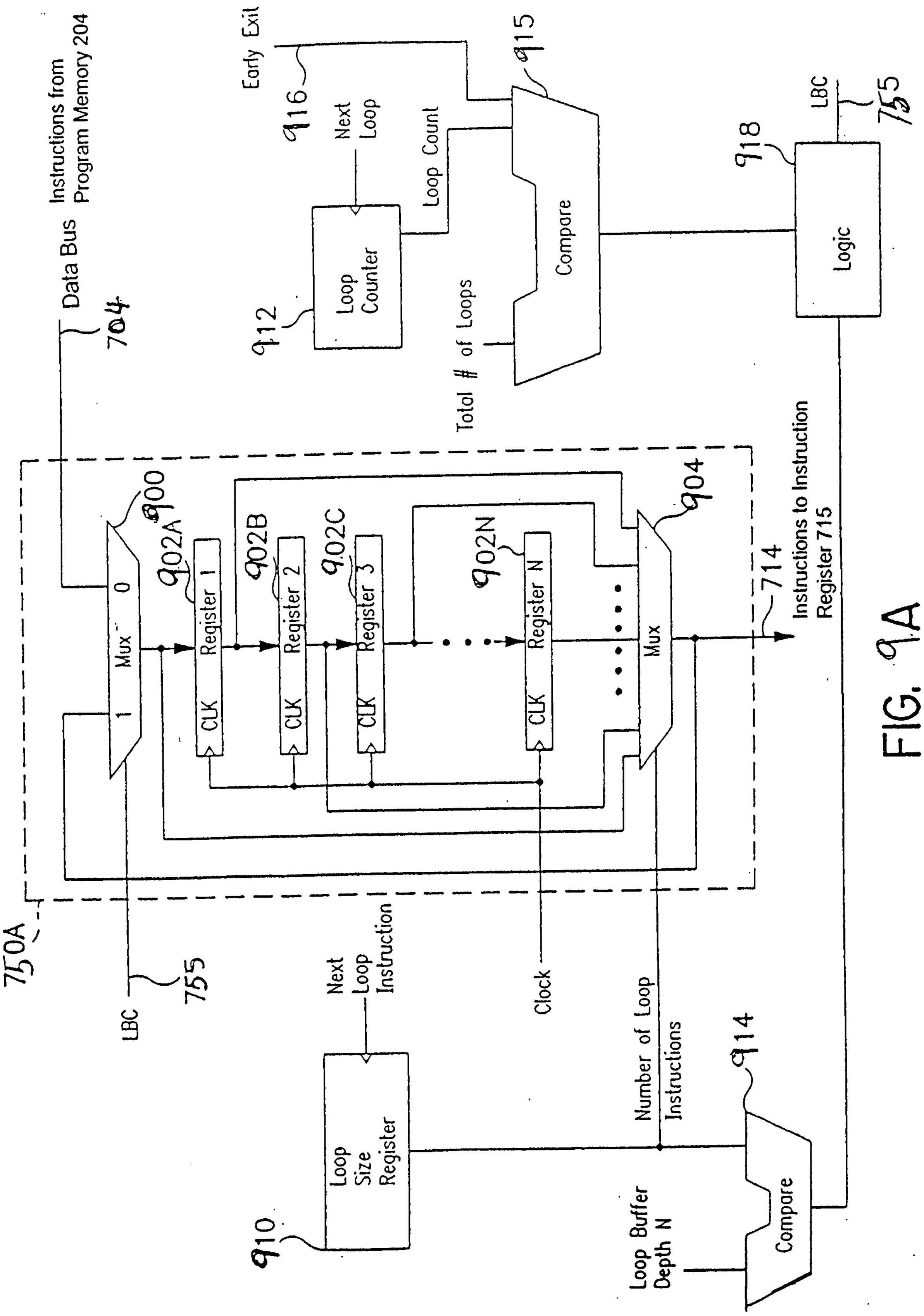

Instructions from Program Memory 204
Data Bus
704
Early Exit
916
915
Next Loop
Loop Count
912
Loop Counter
Total # of Loops
Compare
918
Logic
LBC
755
750A
900
Mux
0
1
902A
Register 1
CLK
902B
Register 2
CLK
902C
Register 3
CLK
902N
Register N
CLK
904
Mux
714
Instructions to Instruction Register 715
LBC
755
Next Loop Instruction
Clock
Number of Loop Instructions
910
Loop Size Register
914
Compare
Loop Buffer Depth N

FIG. 9A

Blakely, Sokoloff, Taylor & Zafman LLP (714) 557-3800
Title: Methods for Power Reduction in a Digital Signal Processor Integrated Circuit
1st Named Inventor: Ruban Kanapathippillai
Application No.: 10/645,366 Docket No.: 42P14037D
Sheet: 20 of 64

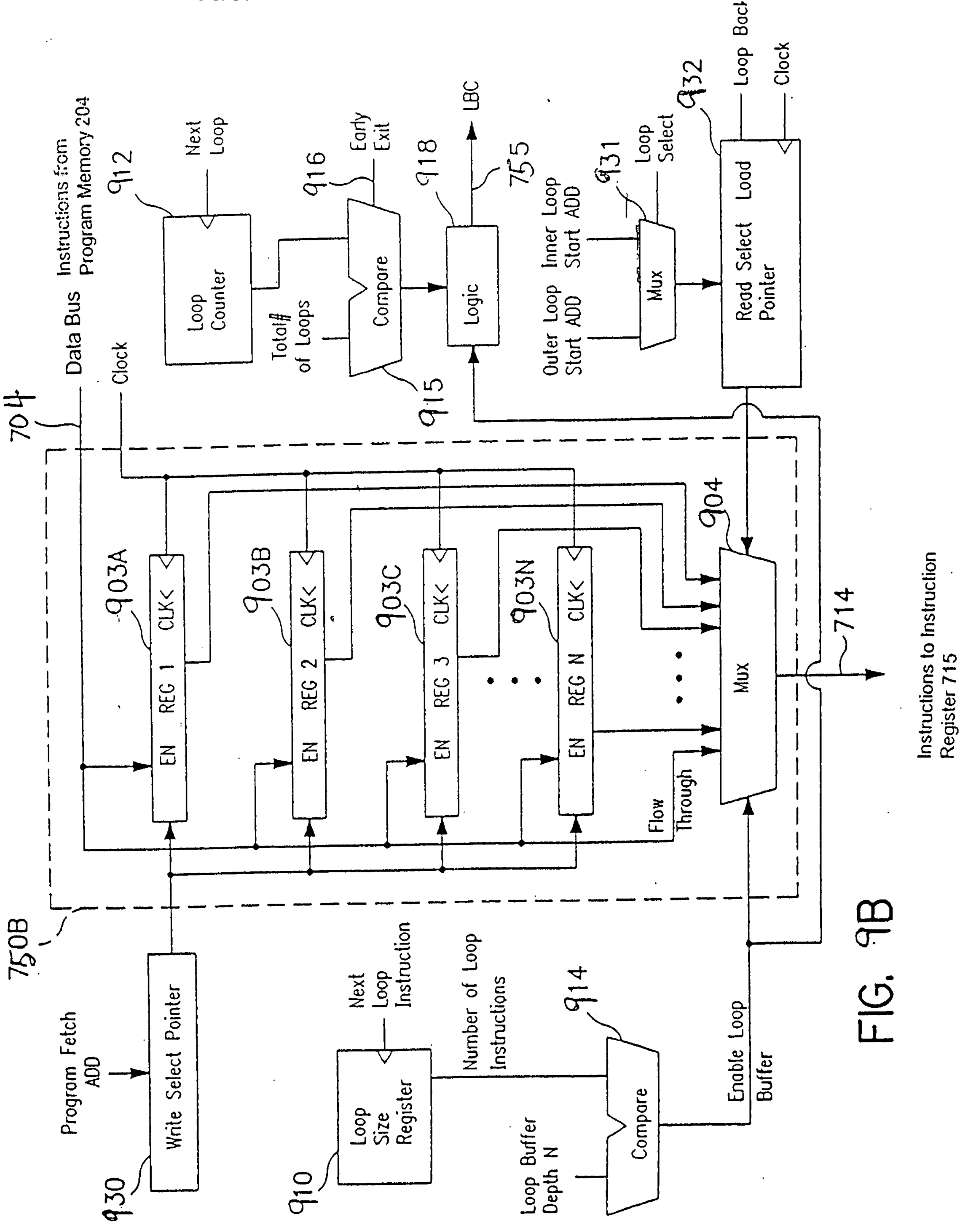


FIG. 9B

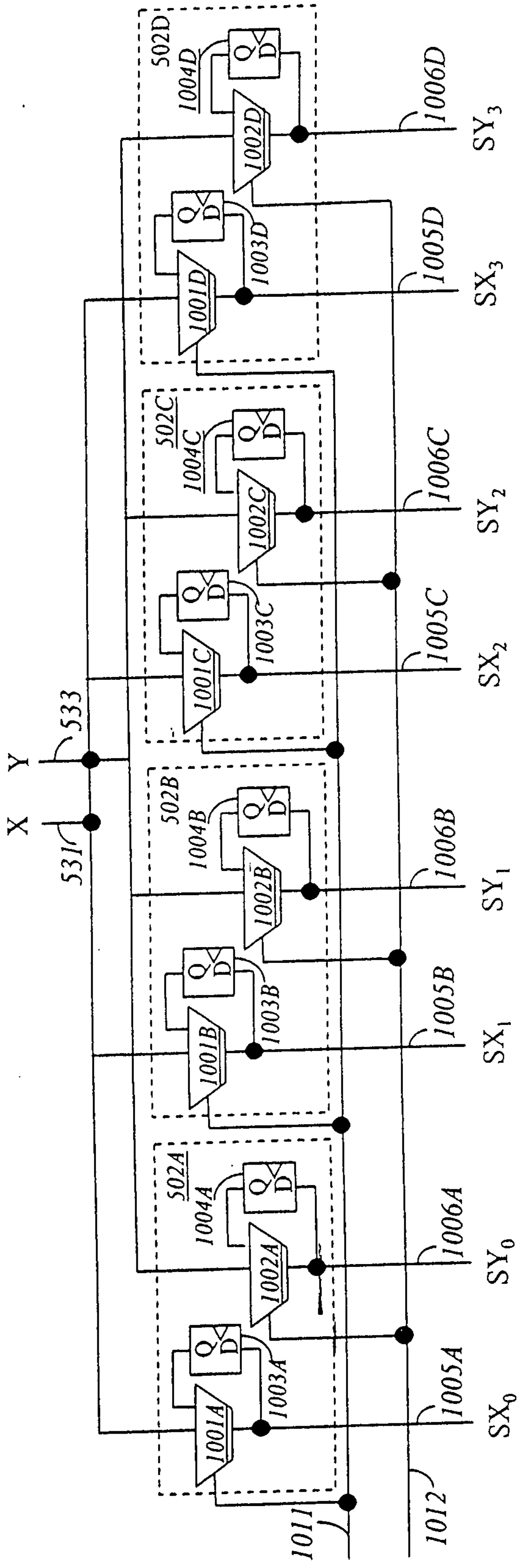

X
531
Y
533
502A
502B
502C
502D
1001A
1001B
1001C
1001D
1002A
1002B
1002C
1002D
1003A
1003B
1003C
1003D
1004A
1004B
1004C
1004D
1005A
1005B
1005C
1005D
1006A
1006B
1006C
1006D
Q
D
SX_0
SY_0
SX_1
SY_1
SX_2
SY_2
SX_3
SY_3
1011
1012

FIG. 10

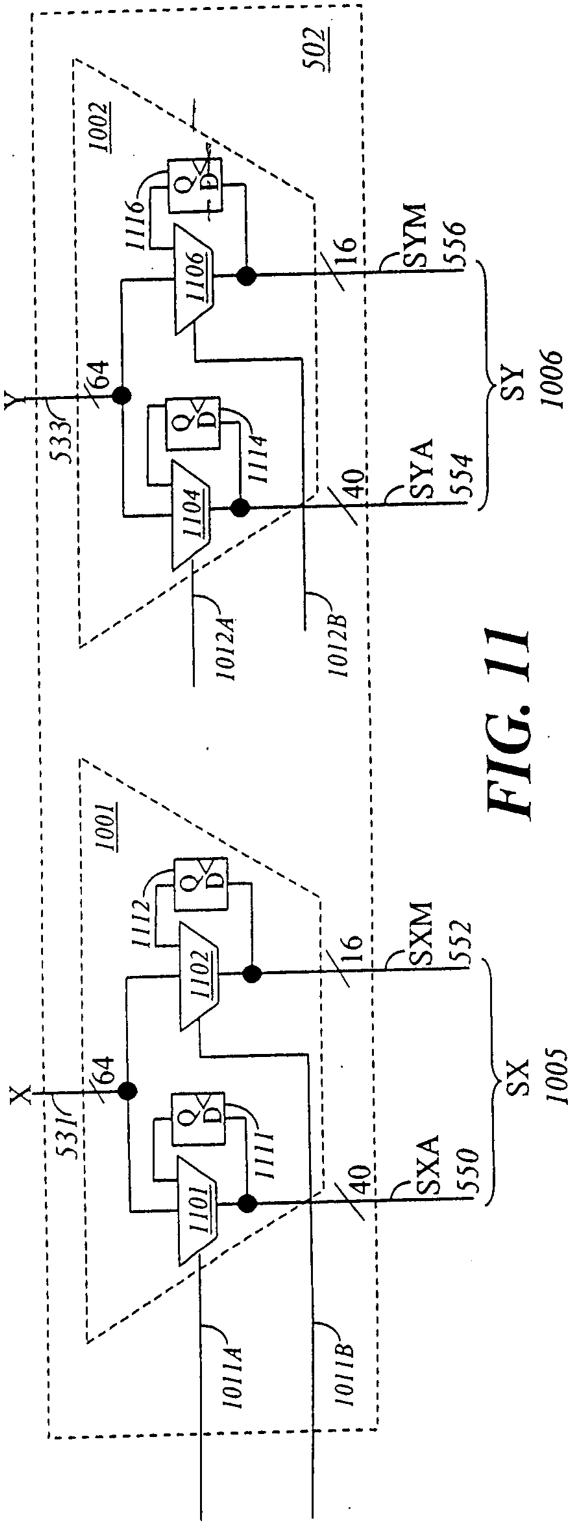

502
1001
1002
X
531
64
Y
533
1101
1102
1104
1106
1111
1112
1114
1116
Q
D
40
16
SXA
550
SXM
552
SX
1005
SYA
554
SYM
556
SY
1006
1011A
1011B
1012A
1012B

FIG. 11

FIG. 12A

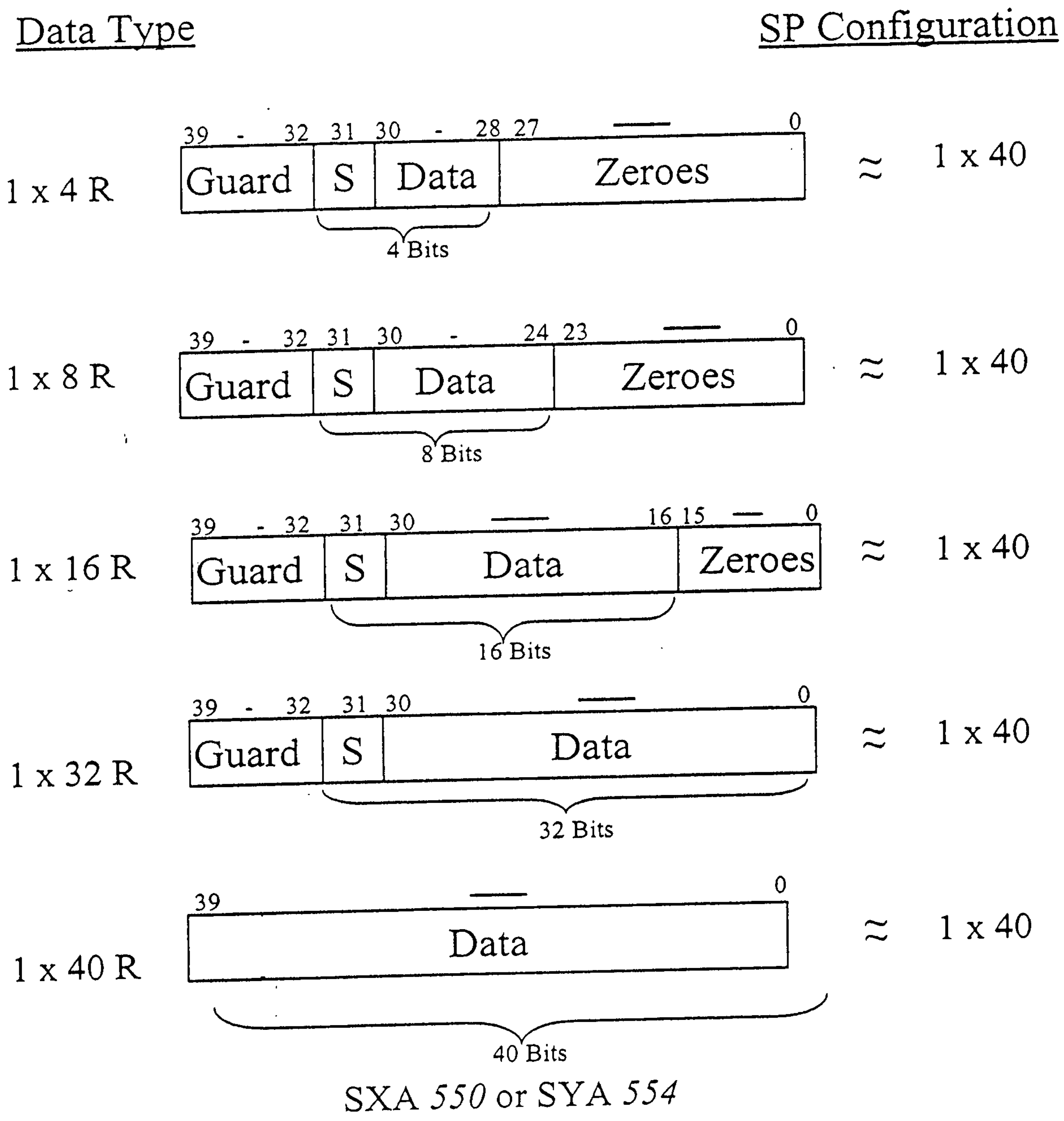

Data Type
SP Configuration
1 x 4 R
39 - 32 31 30 - 28 27 — 0
Guard
S
Data
Zeroes
4 Bits
≈ 1 x 40
1 x 8 R
39 - 32 31 30 - 24 23 — 0
Guard
S
Data
Zeroes
8 Bits
≈ 1 x 40
1 x 16 R
39 - 32 31 30 — 16 15 — 0
Guard
S
Data
Zeroes
16 Bits
≈ 1 x 40
1 x 32 R
39 - 32 31 30 — 0
Guard
S
Data
32 Bits
≈ 1 x 40
1 x 40 R
39 — 0
Data
40 Bits
≈ 1 x 40
SXA 550 or SYA 554

FIG. 12B

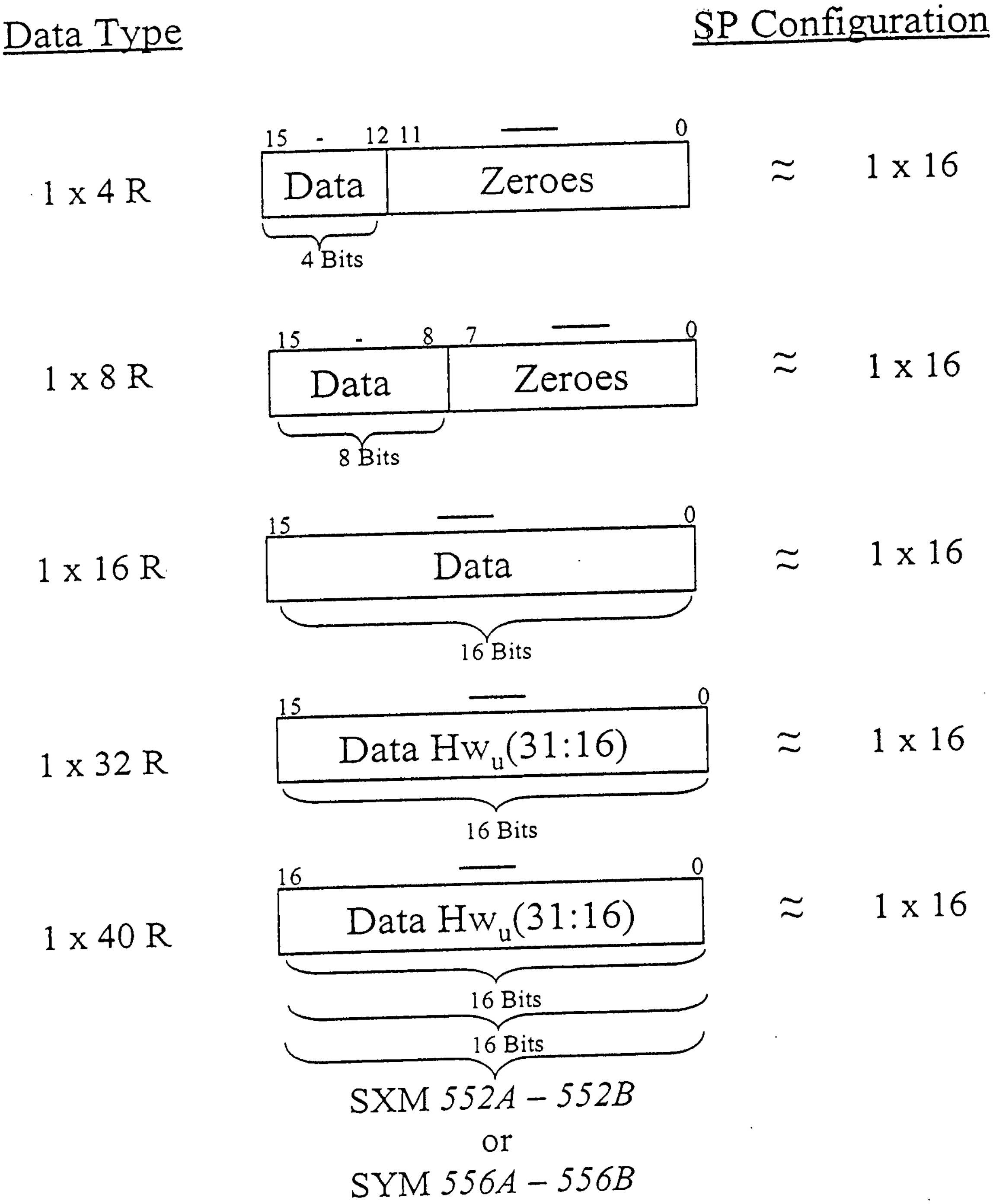

Data Type
SP Configuration
1 x 4 R
15 - 12 11 — 0
Data
Zeroes
4 Bits
≈
1 x 16
1 x 8 R
15 - 8 7 — 0
Data
Zeroes
8 Bits
≈
1 x 16
1 x 16 R
15 — 0
Data
16 Bits
≈
1 x 16
1 x 32 R
15 — 0
Data Hwu(31:16)
16 Bits
≈
1 x 16
1 x 40 R
16 — 0
Data Hwu(31:16)
16 Bits
16 Bits
≈
1 x 16
SXM 552A – 552B
or
SYM 556A – 556B

FIG. 12C

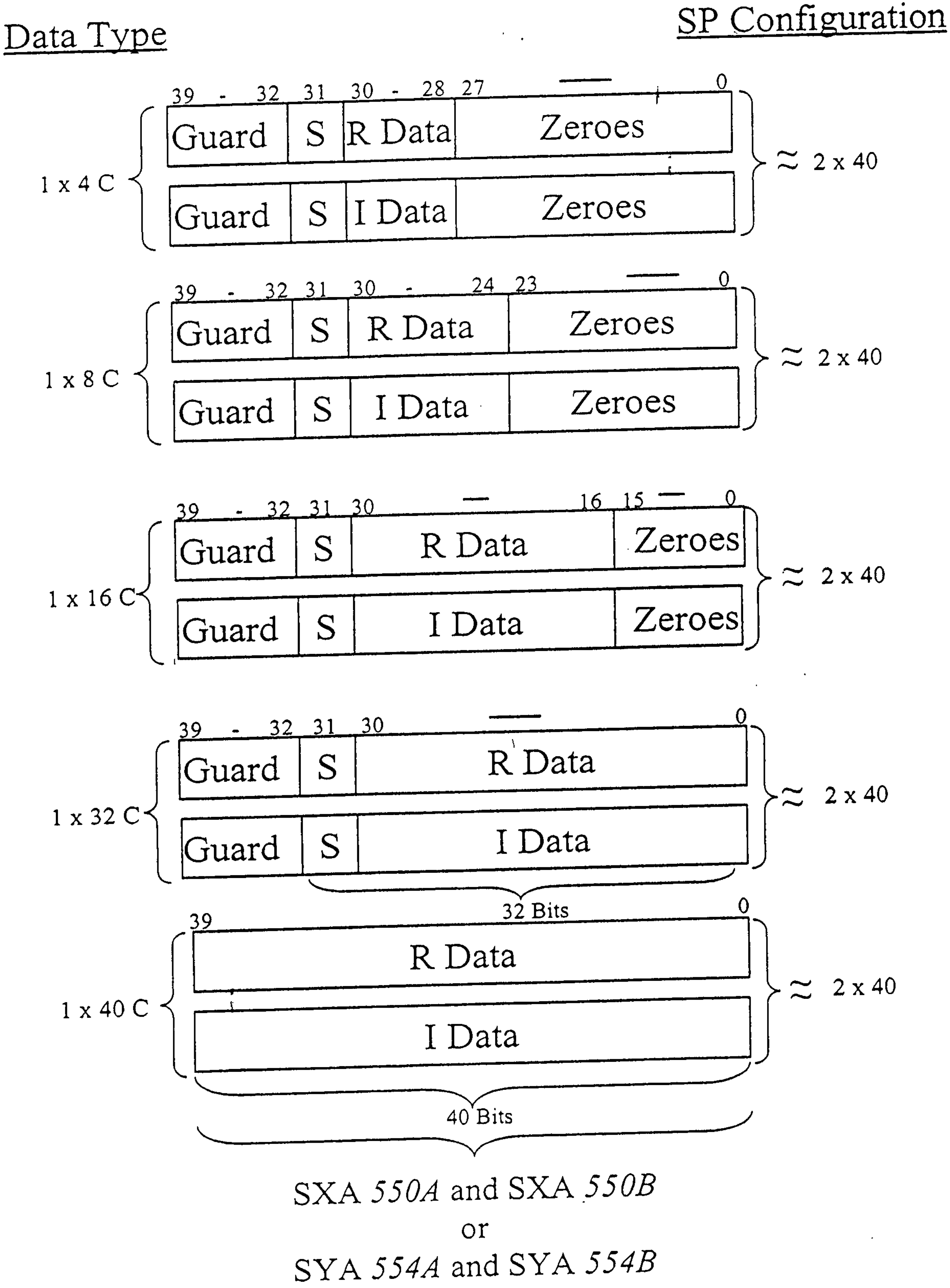

Data Type
SP Configuration
39 - 32 31 30 - 28 27 — 0
Guard
S
R Data
Zeroes
Guard
S
I Data
Zeroes
1 x 4 C
≈ 2 x 40
39 - 32 31 30 - 24 23 — 0
Guard
S
R Data
Zeroes
Guard
S
I Data
Zeroes
1 x 8 C
≈ 2 x 40
39 - 32 31 30 — 16 15 — 0
Guard
S
R Data
Zeroes
Guard
S
I Data
Zeroes
1 x 16 C
≈ 2 x 40
39 - 32 31 30 — 0
Guard
S
R Data
Guard
S
I Data
1 x 32 C
≈ 2 x 40
32 Bits
39
0
R Data
I Data
1 x 40 C
≈ 2 x 40
40 Bits
SXA 550A and SXA 550B
or
SYA 554A and SYA 554B

FIG. 12D

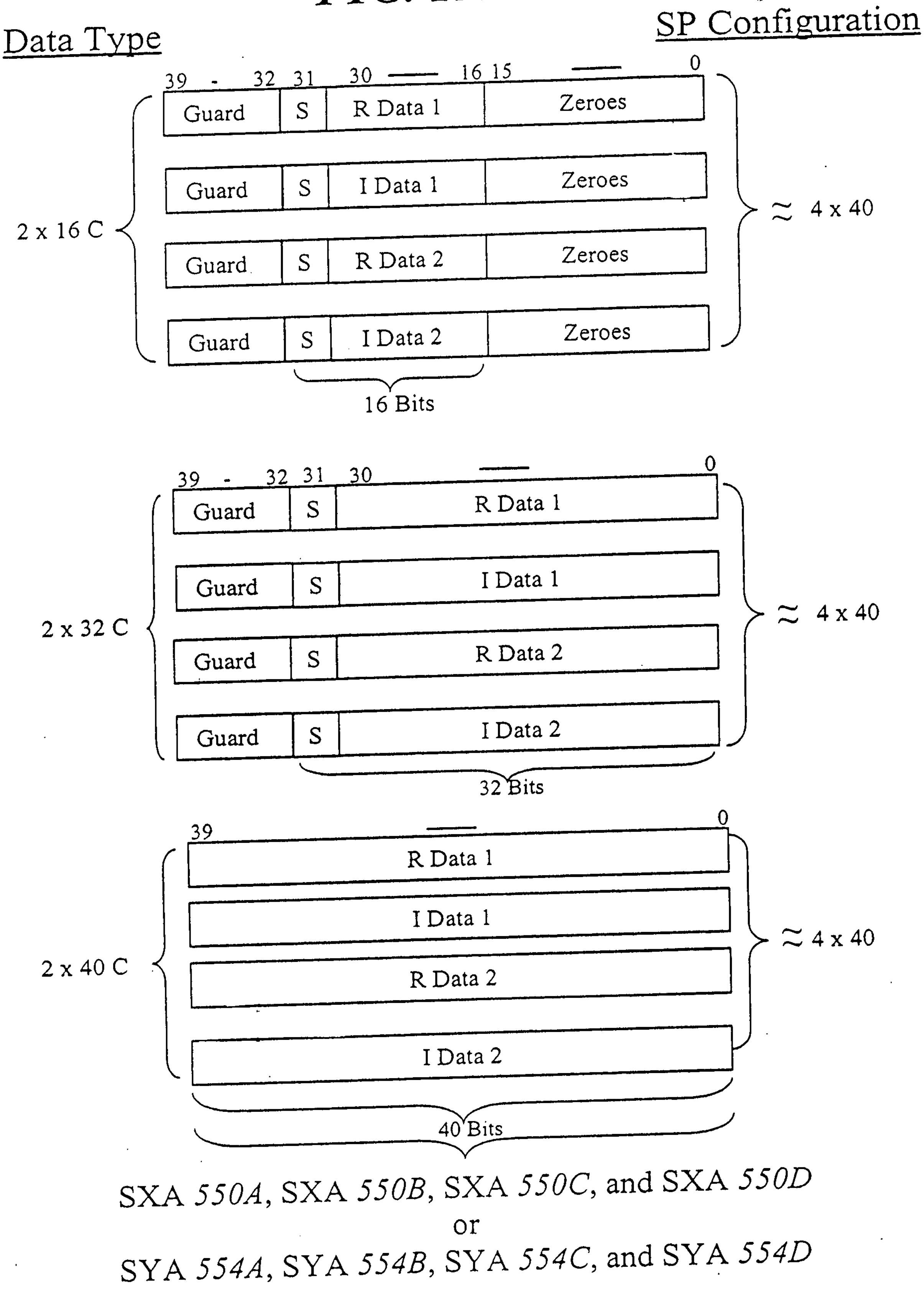

Data Type
SP Configuration
39 - 32 31 30 — 16 15 — 0
Guard
S
R Data 1
Zeroes
I Data 1
R Data 2
I Data 2
2 x 16 C
≈ 4 x 40
16 Bits
39 - 32 31 30 — 0
2 x 32 C
32 Bits
39 — 0
2 x 40 C
40 Bits
SXA 550A, SXA 550B, SXA 550C, and SXA 550D
or
SYA 554A, SYA 554B, SYA 554C, and SYA 554D

FIG. 12E

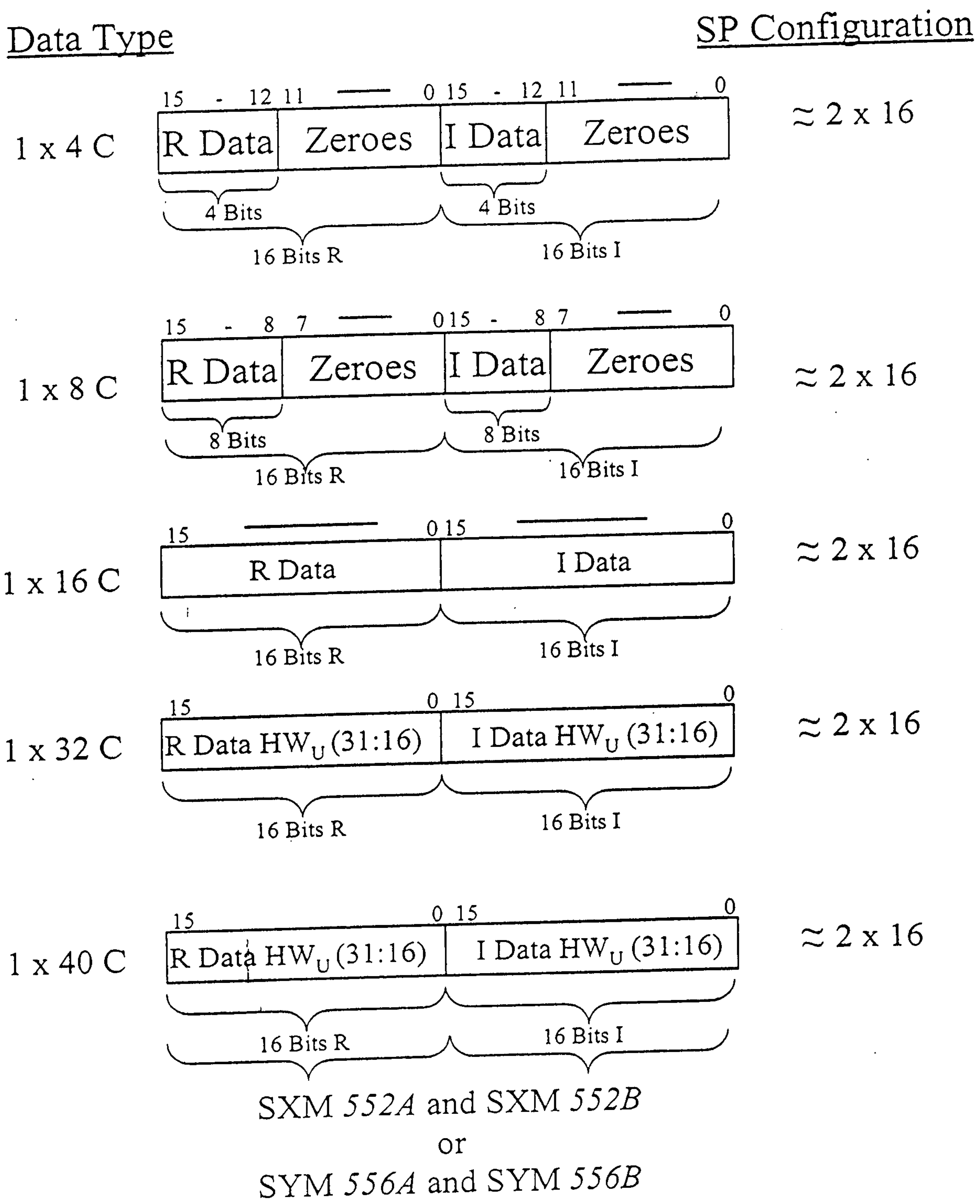
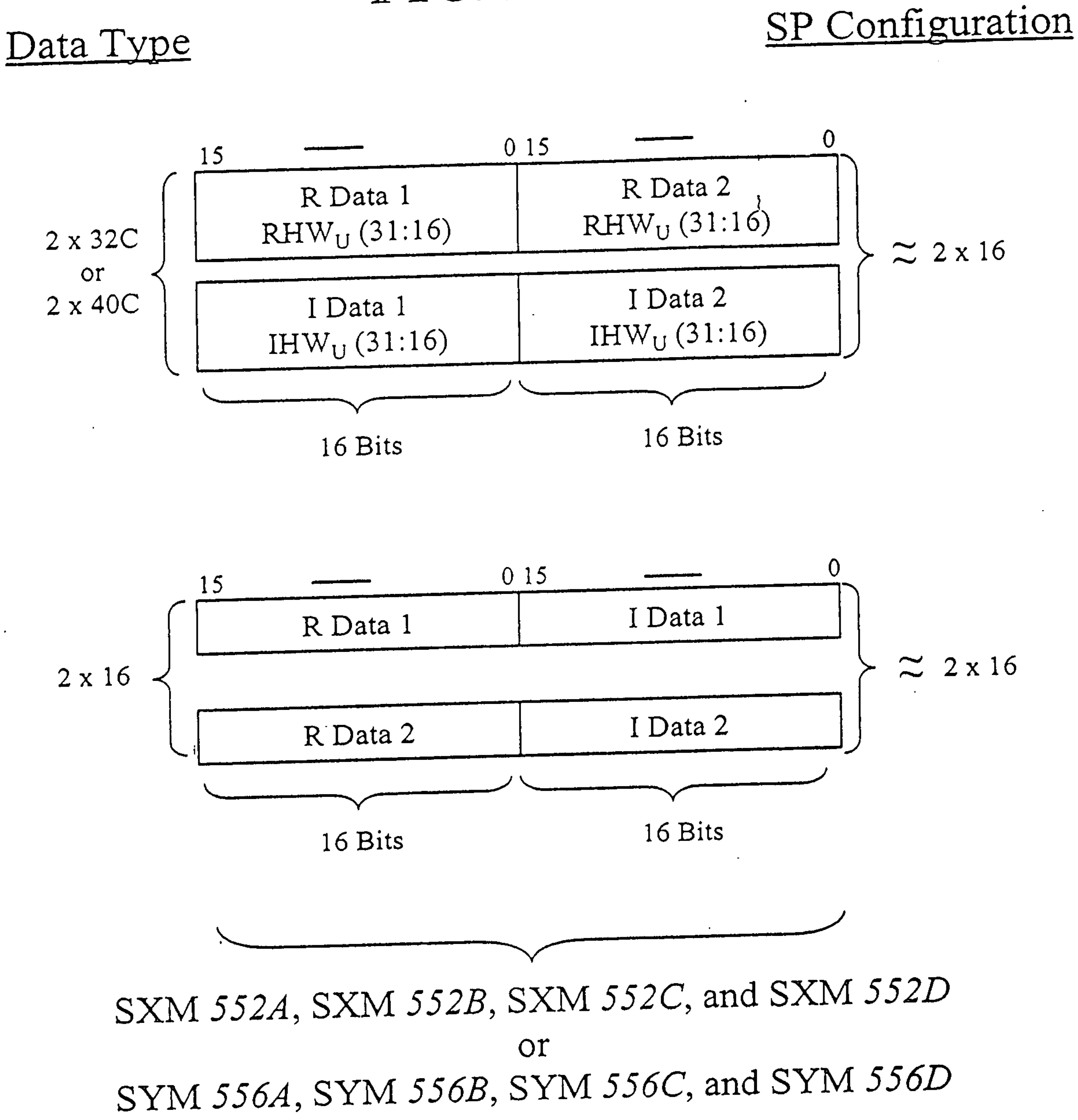

Data Type
SP Configuration
1 x 4 C
15 - 12 11 — 0 15 - 12 11 — 0
R Data
Zeroes
I Data
Zeroes
4 Bits
4 Bits
16 Bits R
16 Bits I
≈ 2 x 16
1 x 8 C
15 - 8 7 — 0 15 - 8 7 — 0
R Data
Zeroes
I Data
Zeroes
8 Bits
8 Bits
16 Bits R
16 Bits I
≈ 2 x 16
1 x 16 C
15 — 0 15 — 0
R Data
I Data
16 Bits R
16 Bits I
≈ 2 x 16
1 x 32 C
15 0 15 0
R Data HW_U (31:16)
I Data HW_U (31:16)
16 Bits R
16 Bits I
≈ 2 x 16
1 x 40 C
15 0 15 0
R Data HW_U (31:16)
I Data HW_U (31:16)
16 Bits R
16 Bits I
≈ 2 x 16
SXM *552A* and SXM *552B*
or
SYM *556A* and SYM *556B*

FIG. 12F


Data Type
SP Configuration
15
0 15
0
R Data 1
RHW$_U$ (31:16)
R Data 2
RHW$_U$ (31:16)
I Data 1
IHW$_U$ (31:16)
I Data 2
IHW$_U$ (31:16)
2 x 32C
or
2 x 40C
≈ 2 x 16
16 Bits
16 Bits
15
0 15
0
R Data 1
I Data 1
R Data 2
I Data 2
2 x 16
≈ 2 x 16
16 Bits
16 Bits
SXM *552A*, SXM *552B*, SXM *552C*, and SXM *552D*
or
SYM *556A*, SYM *556B*, SYM *556C*, and SYM *556D*

Operand 1 Data Type:	N_1 x S_1 R
Operand 2 Data Type:	N_2 x S_2 R
Type Matching R:	Max (N_1 or N_2) x Max (S_1 or S_2) R

Fig. 13A

Operand 1 Data Type:	N_1 x S_1 C
Operand 2 Data Type:	N_2 x S_2 C
Type Matching C:	Max (N_1 or N_2) x Max (S_1 or S_2) C

Fig. 13B

Operand 1 Data Type:	N_1 x S_1 R
Operand 2 Data Type:	N_2 x S_2 C
Type Matching R+C:	Max (N_1 or N_2) x Max (S_1 or S_2) C

Fig. 13C

	1x16 real	2x16 real	1x16 cmpx	4x16 real	2x16 cmpx	1x32 real	2x32 real	1x32 cmpx	4x32 real	2x32 cmpx	1x40 real	2x40 real	1x40 real	4x40 real	2x40 cmpx
1x16 real	1 unit	2 unit	2 unit	4 unit	4 unit	2 unit	4 unit	4 unit							
2x16 real	2 unit	2 unit													
1x16 cmpx	2 unit		4 unit												
4x16 real	4 unit			4 unit											
2x16 cmpx	4 unit														
1x32 real	2 unit														
2x32 real	4 unit														
1x32 cmpx	4 unit														
4x32 real															
2x32 cmpx															
1x40 real															
2x40 real															
1x40 real															
4x40 real															
2x40 real															

FIG. 14

	1x16 real	2x16 real	1x16 cmpx	4x16 real	2x16 cmpx	1x32 real	2x32 real	1x32 cmpx	4x32 real	2x32 cmpx	1x40 real	2x40 real	1x40 real	4x40 real	2x40 cmpx
1x16 real	1 unit	2 unit		4 unit		1 unit	4 unit		4 unit		1 unit	2 unit		4 unit	
2x16 real	2 unit	2 unit				2 unit	2 unit					2 unit			
1x16 cmpx															
4x16 real	4 unit			4 unit		4 unit			4 unit					4 unit	
2x16 cmpx															
1x32 real	1 unit	2 unit		4 unit		1 unit	2 unit		4 unit		1 unit	2 unit		4 unit	
2x32 real	4 unit	2 unit				2 unit	2 unit					2 unit			
1x32 cmpx															
4x32 real	4 unit			4 unit		4 unit			4 unit		4 unit			4 unit	
2x32 cmpx															
1x40 real	1 unit					1 unit			4 unit		1 unit				
2x40 real	2 unit	2 unit				2 unit	2 unit					2 unit			
1x40 real															
4x40 real	4 unit		4 unit			4 unit			4 unit					4 unit	
2x40 real															

FIG. 15A

	1x16 real	2x16 real	1x16 cmpx	4x16 real	2x16 cmpx	1x32 real	2x32 real	1x32 cmpx	4x32 real	2x32 cmpx	1x40 real	2x40 real	1x40 real	4x40 real	2x40 cmpx
1x16 real	1 unit	2 unit	2 unit	4 unit	4 unit	1 unit	2 unit	2 unit	4 unit	4 unit	1 unit	2 unit	2 unit	4 unit	4 unit
2x16 real	2 unit	2 unit				2 unit	2 unit					2 unit			
1x16 cmpx	2 unit		2 unit					2 unit			2 unit		2 unit		
4x16 real	4 unit			1 unit		4 unit			4 unit					4 unit	
2x16 cmpx	4 unit				4 unit					4 unit					4 unit
1x32 real	1 unit	2 unit				1 unit	2 unit	2 unit	4 unit		1 unit	2 unit	2 unit	4 unit	
2x32 real	2 unit	2 unit				2 unit	2 unit					2 unit			
1x32 cmpx	2 unit		2 unit			2 unit		2 unit			2 unit		2 unit		
4x32 real	4 unit			4 unit		4 unit			4 unit		4 unit			4 unit	
2x32 cmpx	4 unit				4 unit					4 unit					4 unit
1x40 real	1 unit		2 unit			1 unit		2 unit	4 unit		1 unit	2 unit		4 unit	
2x40 real	2 unit	2 unit				2 unit	2 unit				2 unit	2 unit			
1x40 real	2 unit		2 unit			2 unit		2 unit					2 unit		
4x40 real	4 unit			4 unit		4 unit			4 unit		4 unit			4 unit	
2x40 real	4 unit				4 unit					4 unit					4 unit

FIG. 15B

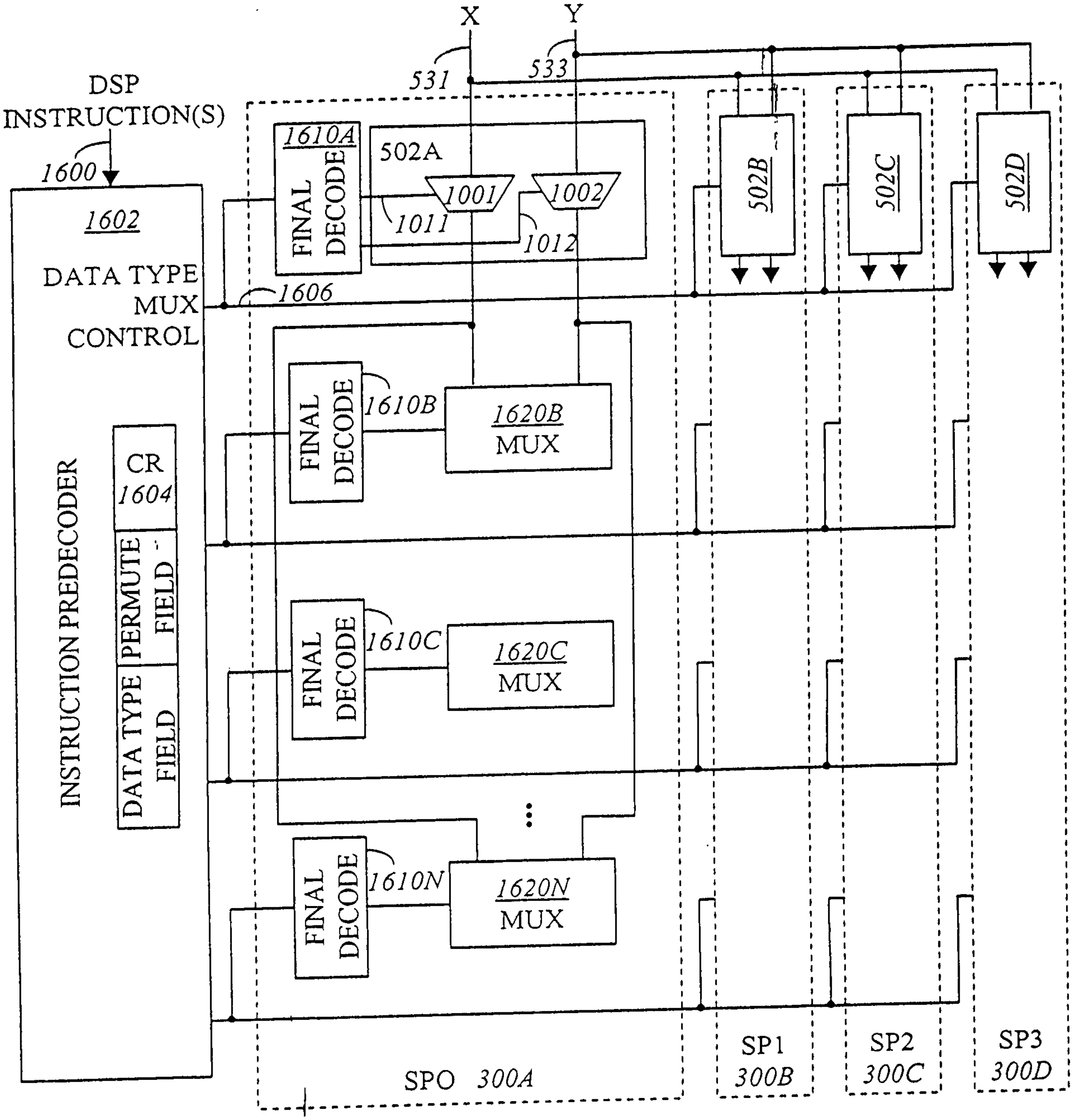

X
Y
531
533
DSP INSTRUCTION(S)
1600
1602
DATA TYPE MUX CONTROL
INSTRUCTION PREDECODER
CR 1604
PERMUTE FIELD
DATA TYPE FIELD
1606
1610A
FINAL DECODE
502A
1001
1002
1011
1012
502B
502C
502D
FINAL DECODE
1610B
1620B MUX
FINAL DECODE
1610C
1620C MUX
FINAL DECODE
1610N
1620N MUX
SPO 300A
SP1 300B
SP2 300C
SP3 300D

FIG. 16

Data Type: N x S(R/C)

FIG. 17

(rX): Memory Address Registers

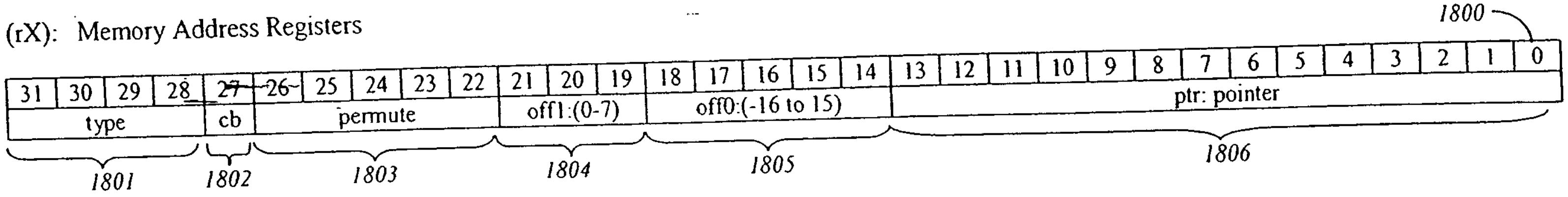


FIG. 18

Data Type 1801

0000:	1x16 real
0001:	2x16 real
0010:	1x16 complex
0011:	4x16 real
0100:	1x32 real
0101:	2x32 real
0110:	1x32 complex
0111:	2x16 complex
1000:	4x32 real
1001:	2x32 complex
1010:	1x40 real
1011:	2x40 real
1100:	1x40 complex
1101:	4x40 real (only for local add unit operations)
1110:	2x40 complex (only for local add unit operations)
1111:	Reserved

FIG. 19

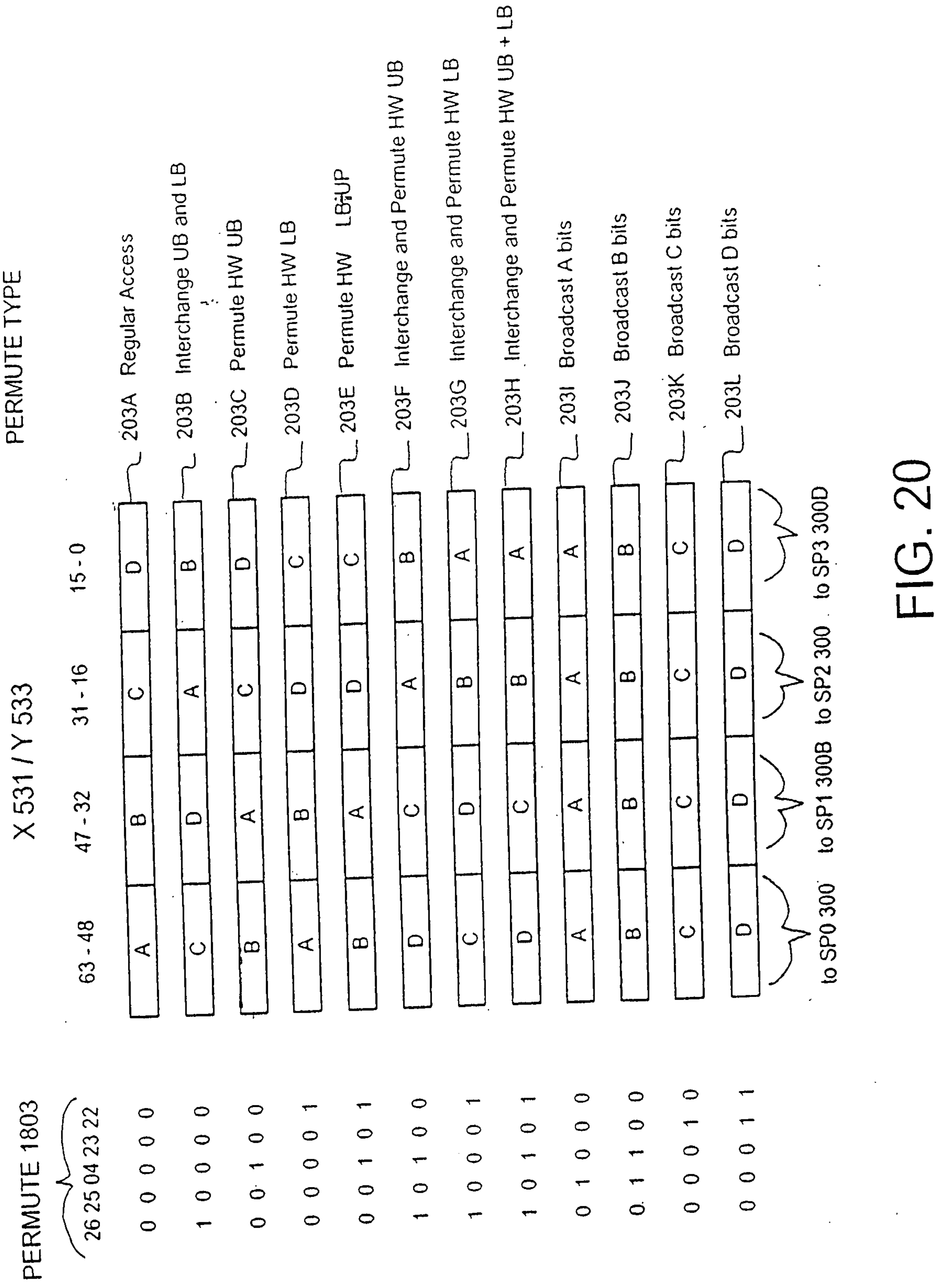

PERMUTE 1803
26 25 04 23 22
X 531 / Y 533
63 - 48
47 - 32
31 - 16
15 - 0
PERMUTE TYPE
0 0 0 0 0
A B C D
203A Regular Access
1 0 0 0 0
C D A B
203B Interchange UB and LB
0 0 1 0 0
B A C D
203C Permute HW UB
0 0 0 0 1
A B D C
203D Permute HW LB
0 0 1 0 1
B A D C
203E Permute HW LB+UP
1 0 1 0 0
D C A B
203F Interchange and Permute HW UB
1 0 0 0 1
C D B A
203G Interchange and Permute HW LB
1 0 1 0 1
D C B A
203H Interchange and Permute HW UB + LB
0 1 0 0 0
A A A A
203I Broadcast A bits
0 1 1 0 0
B B B B
203J Broadcast B bits
0 0 0 1 0
C C C C
203K Broadcast C bits
0 0 0 1 1
D D D D
203L Broadcast D bits
to SP0 300
to SP1 300B
to SP2 300
to SP3 300D

FIG. 20

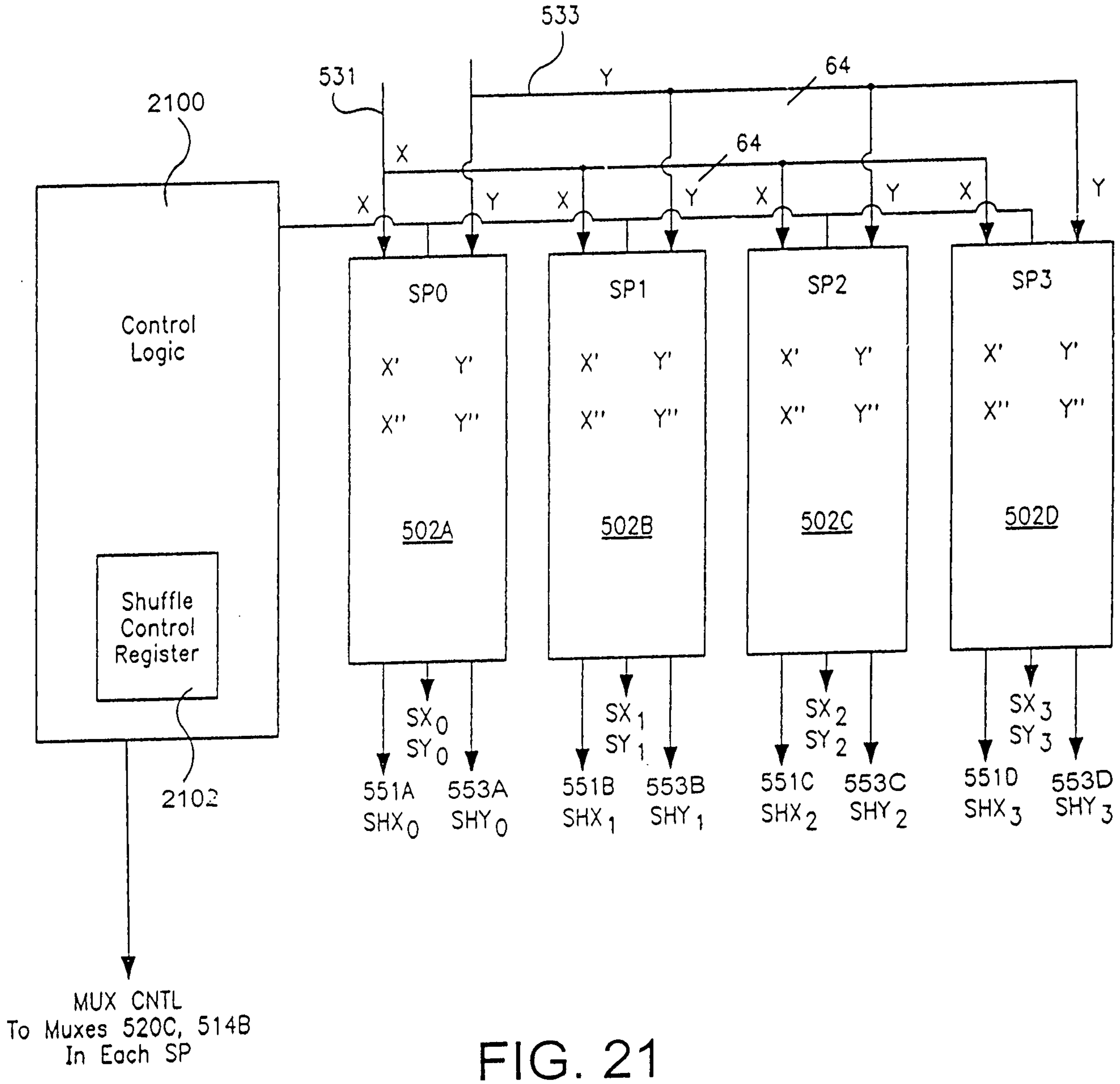

533
531
64
Y
X
64
2100
X
Y
X
Y
X
Y
X
Y
Control Logic
Shuffle Control Register
SP0
X'
Y'
X''
Y''
502A
SP1
X'
Y'
X''
Y''
502B
SP2
X'
Y'
X''
Y''
502C
SP3
X'
Y'
X''
Y''
502D
SX0
SY0
551A SHX0
553A SHY0
SX1
SY1
551B SHX1
553B SHY1
SX2
SY2
551C SHX2
553C SHY2
SX3
SY3
551D SHX3
553D SHY3
2102
MUX CNTL
To Muxes 520C, 514B
In Each SP

FIG. 21

$X' = [\ SX_{10},\ SX_{11},\ SX_{12},\ SX_{13}\ ]$ e.g. $[\ X_0,\ X_1,\ X_2,\ X_3\ ]$

$X'' = [\ SX_{20},\ SX_{21},\ SX_{22},\ SX_{23}\ ]$ e.g. $[\ X_4,\ X_5,\ X_6,\ X_7\ ]$

Where SX_{ab} :S=Source; a=delay; b= SP unit number(e.g. SP3,SP2 SP1,SP0; or termed U3,U2,U1,U0)

$Y' = [\ SY_{10},\ SY_{11},\ SY_{12},\ SY_{13}\ ]$

$Y'' = [\ SY_{20},\ SY_{21},\ SY_{22},\ SY_{23}\ ]$

Where SY_{ab} :S=Source; a=delay; b= SP unit number(e.g. SP3,SP2,SP1, SP0; or termed U3,U1,U0)

FIG. 22A

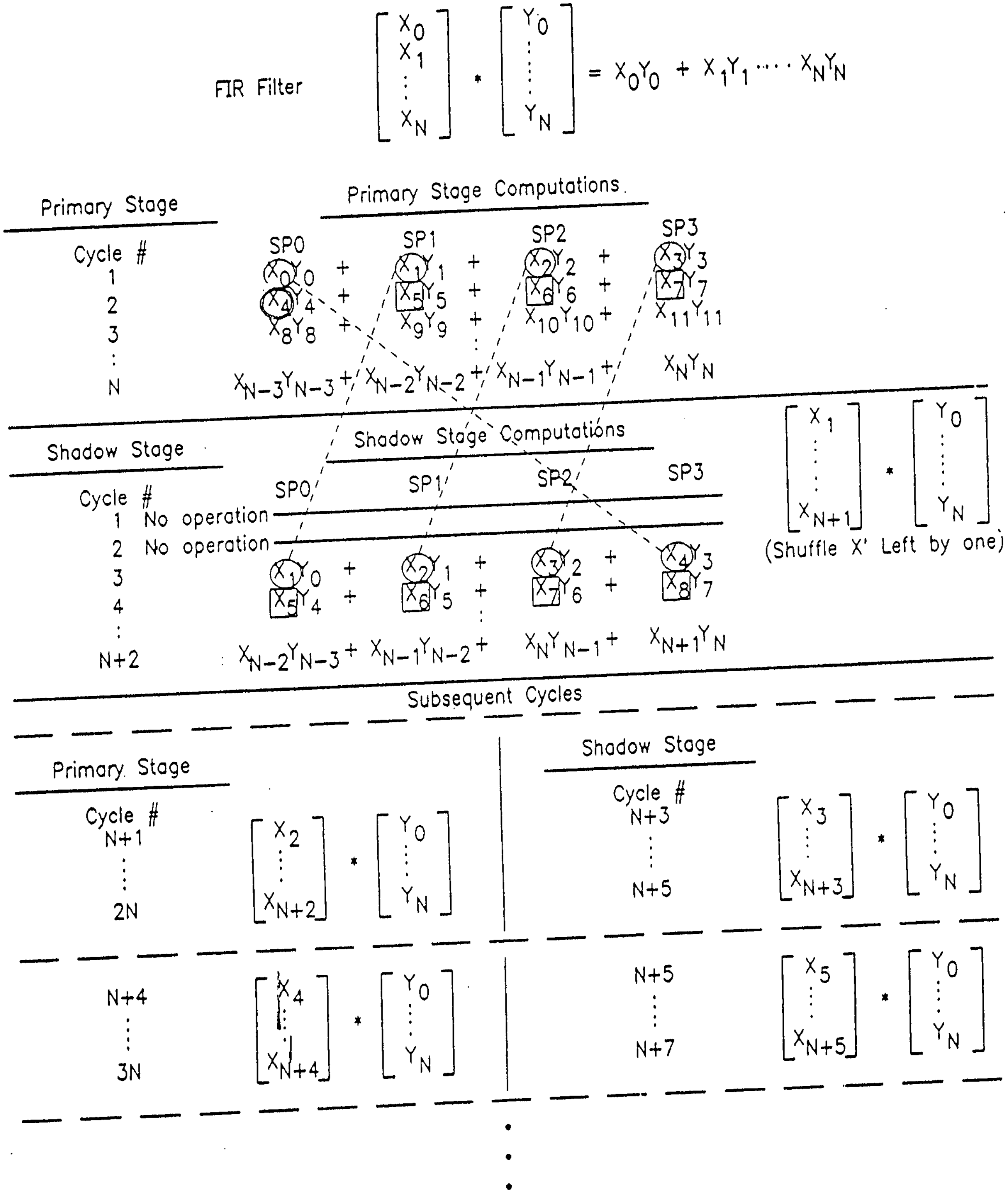

FIR Filter
X_0 X_1 X_N * Y_0 Y_N = $X_0Y_0 + X_1Y_1 \cdots X_NY_N$
Primary Stage
Cycle #
1
2
3
N
Primary Stage Computations
SP0
SP1
SP2
SP3
X_0Y_0 + X_1Y_1 + X_2Y_2 + X_3Y_3
X_4Y_4 + X_5Y_5 + X_6Y_6 + X_7Y_7
X_8Y_8 + X_9Y_9 + $X_{10}Y_{10}$ + $X_{11}Y_{11}$
$X_{N-3}Y_{N-3}$ + $X_{N-2}Y_{N-2}$ + $X_{N-1}Y_{N-1}$ + X_NY_N
Shadow Stage
Cycle #
1 No operation
2 No operation
3
4
N+2
Shadow Stage Computations
SP0
SP1
SP2
SP3
X_1Y_0 + X_2Y_1 + X_3Y_2 + X_4Y_3
X_5Y_4 + X_6Y_5 + X_7Y_6 + X_8Y_7
$X_{N-2}Y_{N-3}$ + $X_{N-1}Y_{N-2}$ + X_NY_{N-1} + $X_{N+1}Y_N$
X_1 X_{N+1} * Y_0 Y_N
(Shuffle X' Left by one)
Subsequent Cycles
Primary Stage
Cycle #
N+1
2N
X_2 X_{N+2} * Y_0 Y_N
Shadow Stage
Cycle #
N+3
N+5
X_3 X_{N+3} * Y_0 Y_N
N+4
3N
X_4 X_{N+4} * Y_0 Y_N
N+5
N+7
X_5 X_{N+5} * Y_0 Y_N

FIG. 22B

shuffle Shuffle Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
u3		u2		u1		u0		u3		u2		u1		u0		u3		u2		u1		u0		u3		u2		u1		u0	
SY2S								SY1S								SX2S								SX1S							

Units are connected to their nearest neighbors for shuffling the sources using the following bit diagram:

00	Unit N+1, SX1=X'	(right)
01	Unit N+1, SX2=X"	(right)
10	Unit N-1, SX1=X'	(left)
11	Unit N-1, SX2=X"	(left)

For example to shift the sources to the left by one:

3	2	1	0	From
2	1	0	3	Into

The bits should be 10101010 ($AA)

FIG. 22C

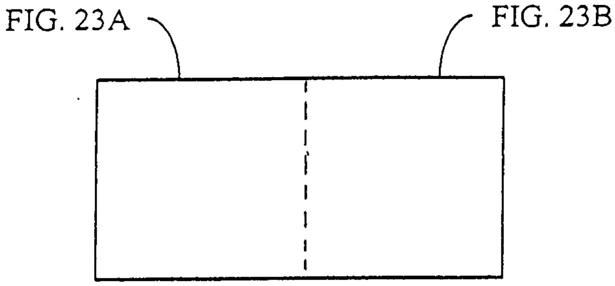


FIG. 23

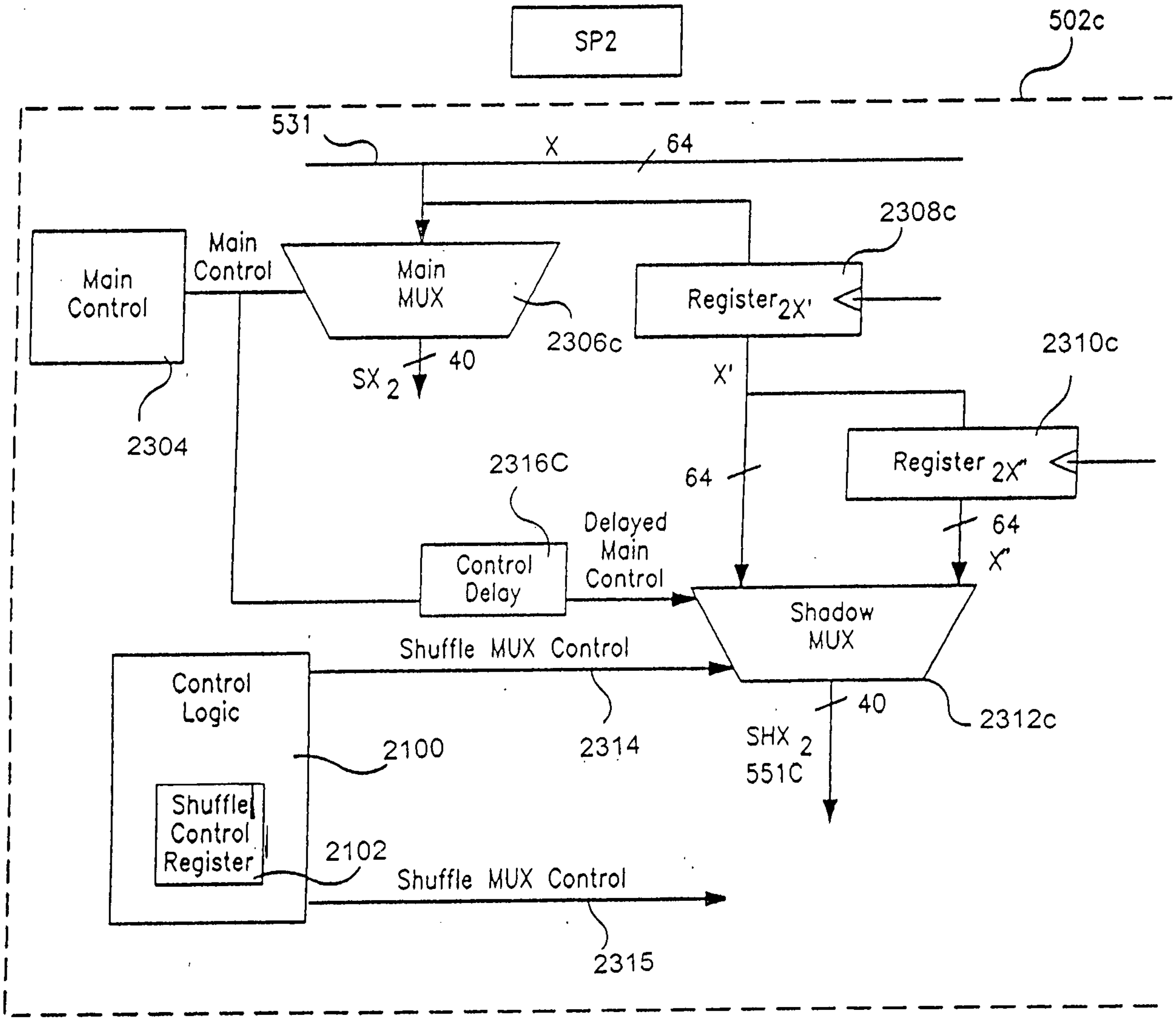


FIG. 23A

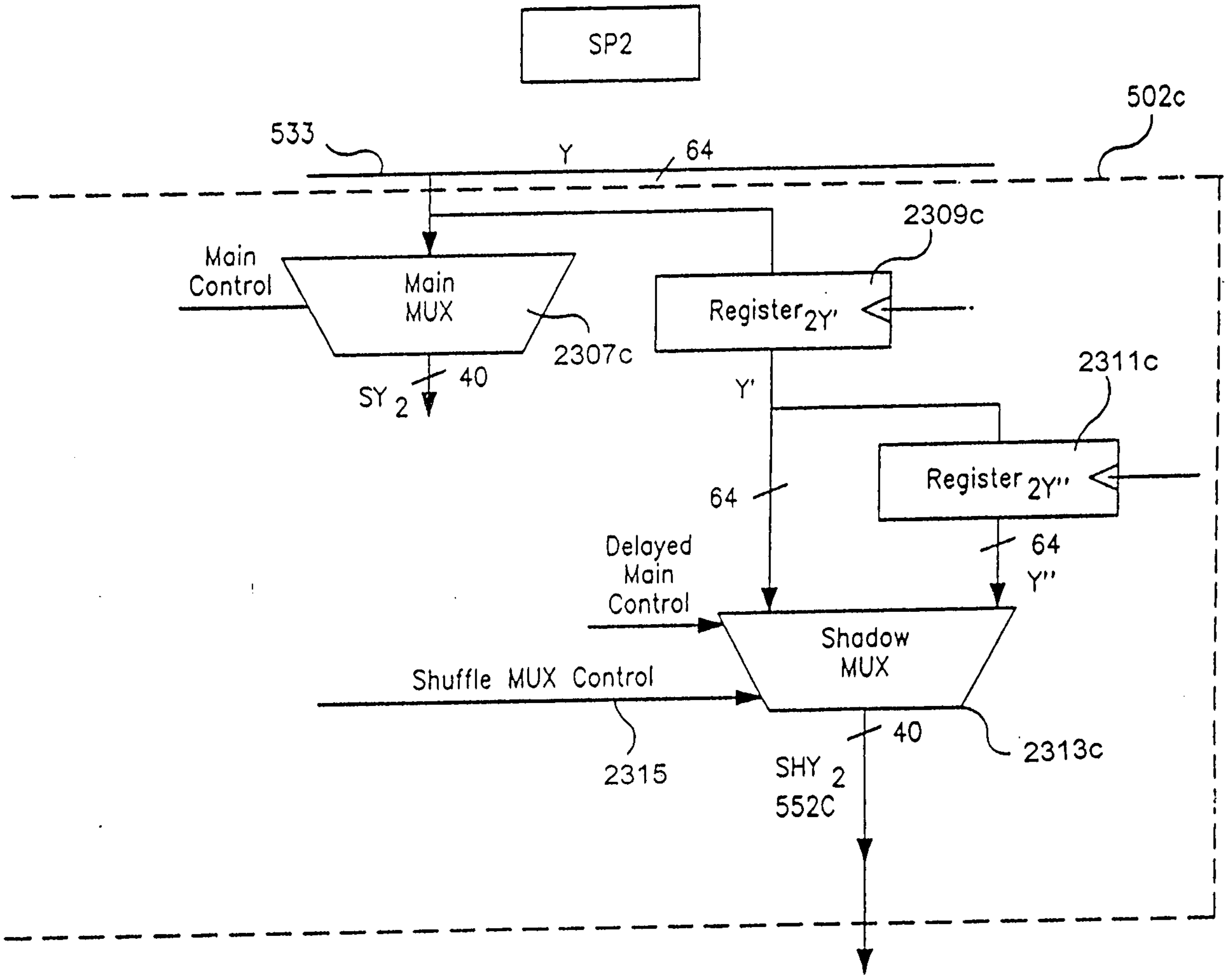
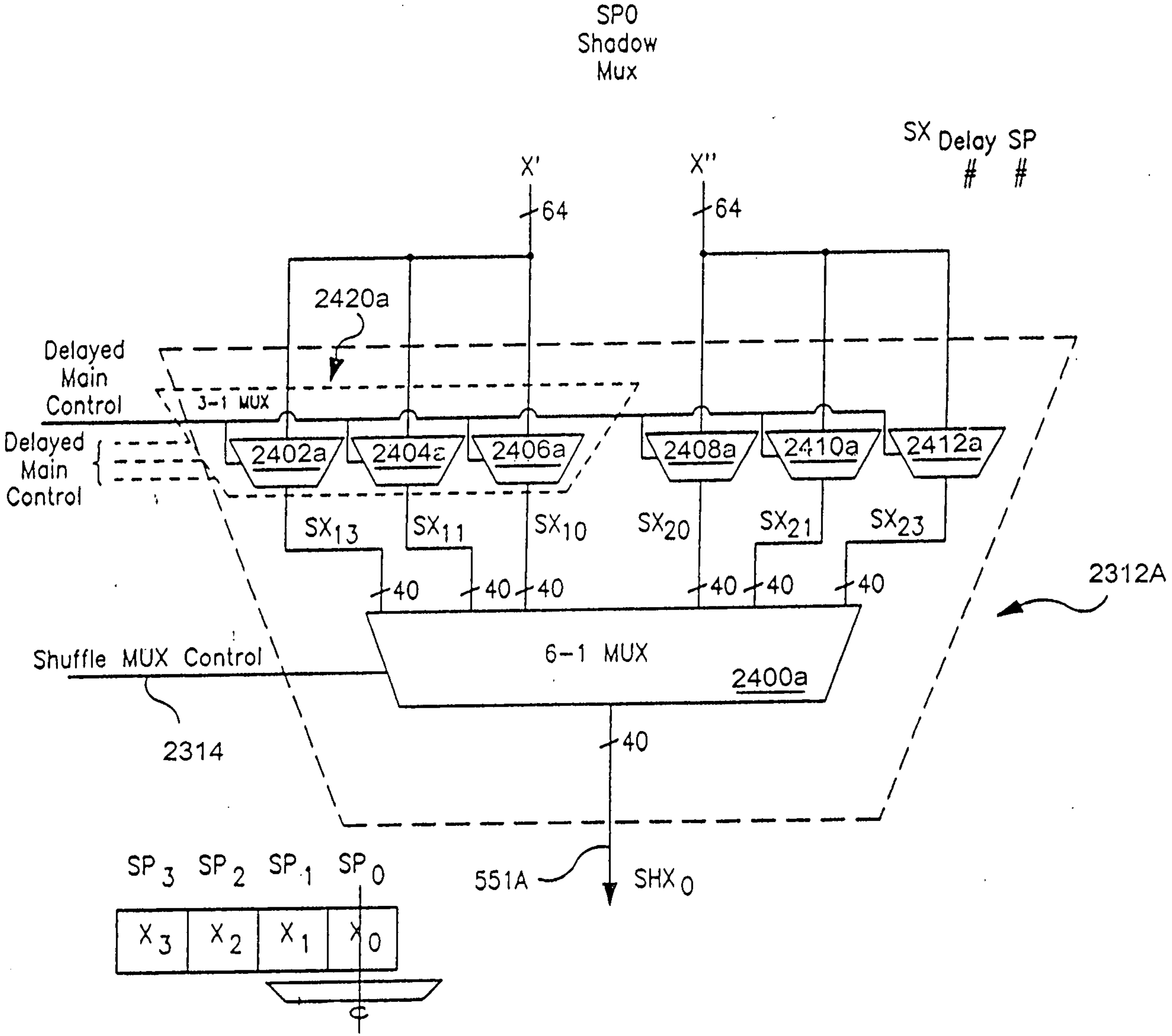

SP2
502c
533
Y
64
2309c
Main Control
Main MUX
2307c
Register 2Y'
40
SY 2
Y'
2311c
64
Register 2Y''
64
Y''
Delayed Main Control
Shadow MUX
Shuffle MUX Control
2315
40
SHY 2
552C
2313c

FIG. 23B



$X_0 = SX_{10}, SX_{20}$

$X_1 = SX_{11}, SX_{21}$

$X_3 = SX_{13}, SX_{23}$

FIG. 24A

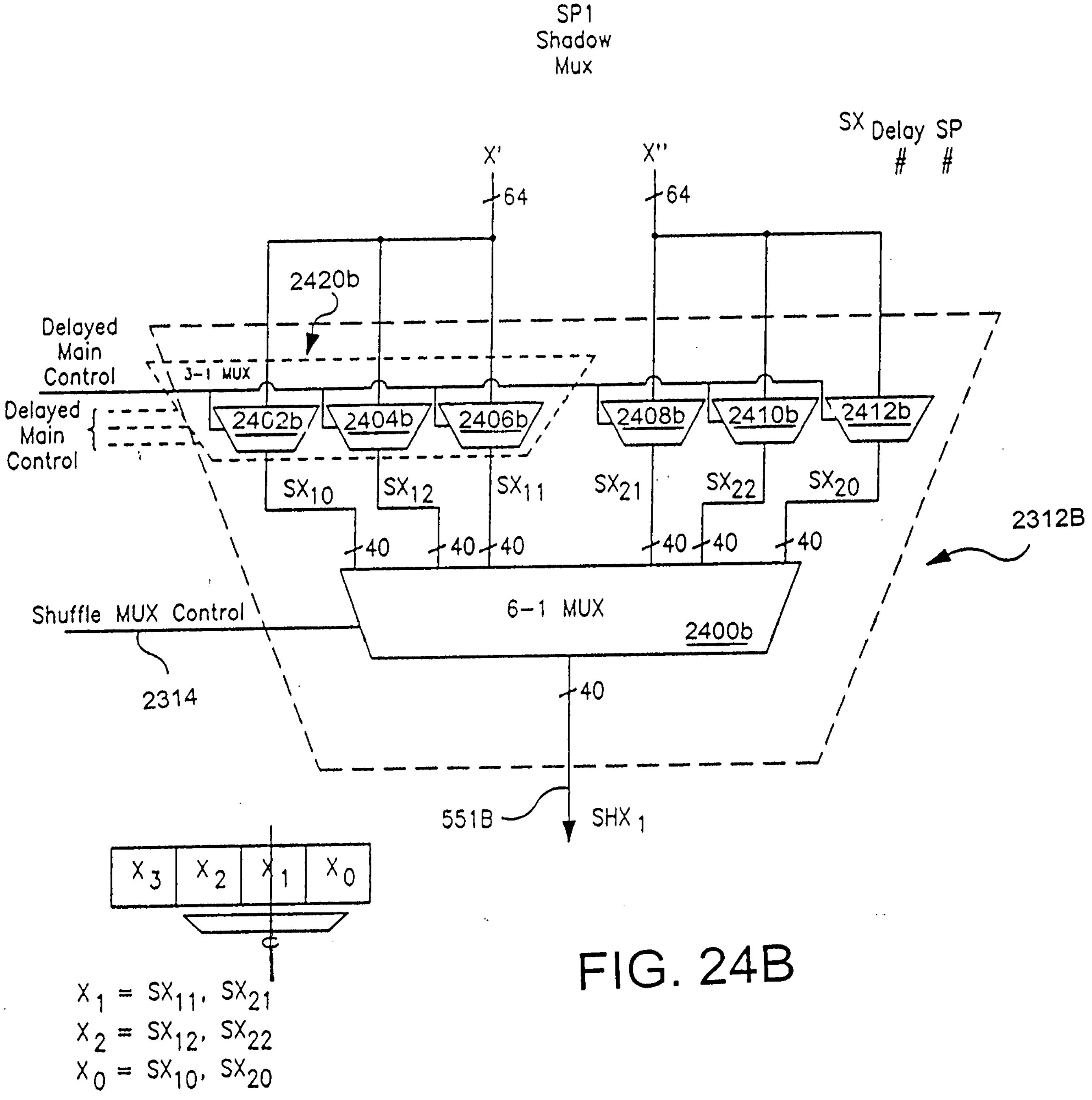

SP1
Shadow
Mux
SX Delay SP
# #
X'
64
X''
64
2420b
Delayed
Main
Control
3-1 MUX
Delayed
Main
Control
2402b
2404b
2406b
2408b
2410b
2412b
SX10
SX12
SX11
SX21
SX22
SX20
40
40
40
40
40
40
6-1 MUX
2400b
Shuffle MUX Control
2314
2312B
40
551B
SHX 1
X3
X2
X1
X0
X1 = SX11, SX21
X2 = SX12, SX22
X0 = SX10, SX20

FIG. 24B

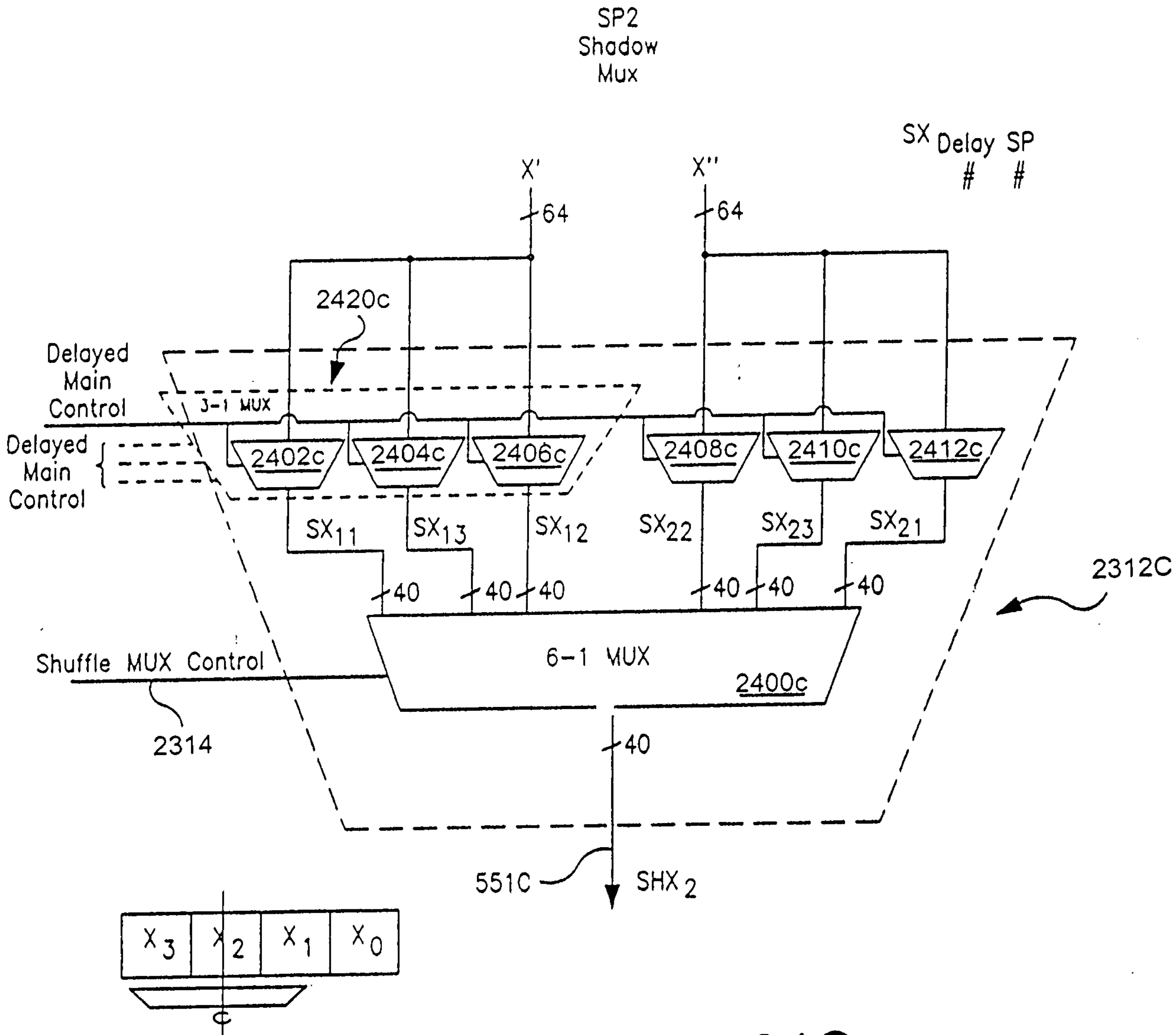
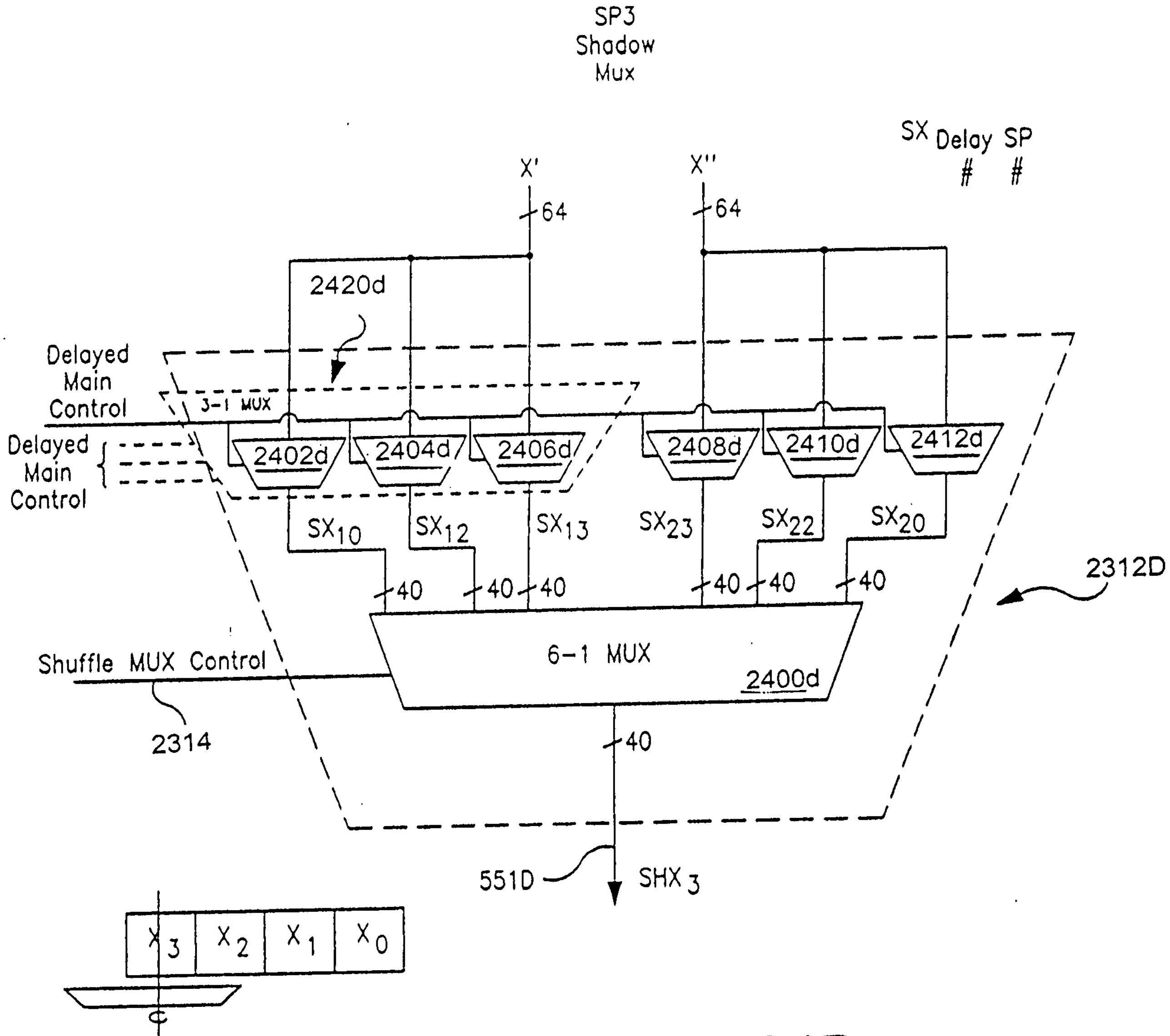


FIG. 24C

$X_2 = SX_{12}, SX_{22}$

$X_3 = SX_{13}, SX_{23}$

$X_1 = SX_{11}, SX_{21}$



X_3 = SX_{13}, SX_{23}
X_0 = SX_{10}, SX_{20}
X_2 = SX_{12}, SX_{22}

FIG. 24D

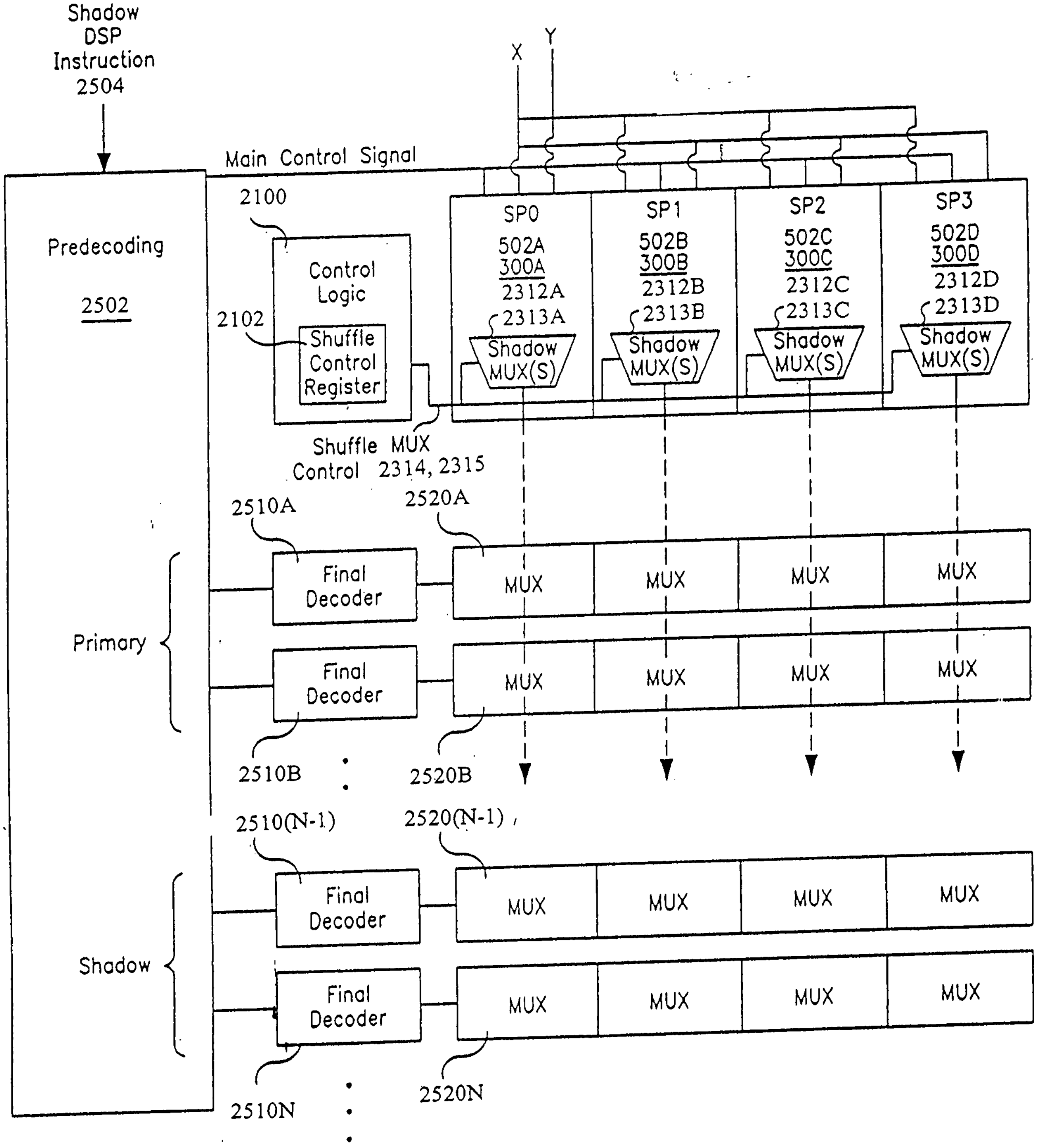

Shadow DSP Instruction 2504
X
Y
Main Control Signal
2100
Predecoding
2502
Control Logic
2102
Shuffle Control Register
SP0
502A
300A
2312A
2313A
Shadow MUX(S)
SP1
502B
300B
2312B
2313B
Shadow MUX(S)
SP2
502C
300C
2312C
2313C
Shadow MUX(S)
SP3
502D
300D
2312D
2313D
Shadow MUX(S)
Shuffle MUX Control 2314, 2315
2510A
2520A
Primary
Final Decoder
MUX
MUX
MUX
MUX
Final Decoder
MUX
MUX
MUX
MUX
2510B
2520B
2510(N-1)
2520(N-1)
Shadow
Final Decoder
MUX
MUX
MUX
MUX
Final Decoder
MUX
MUX
MUX
MUX
2510N
2520N

FIG. 25

Blakely, Sokoloff, Taylor & Zafman LLP (714) 557-3800
Title: Methods for Power Reduction in a Digital Signal Processor Integrated Circuit
1st Named Inventor: Ruban Kanapathippillai
Application No.: 10/645,366 Docket No.: 42P14037D
Sheet: 46 of 64

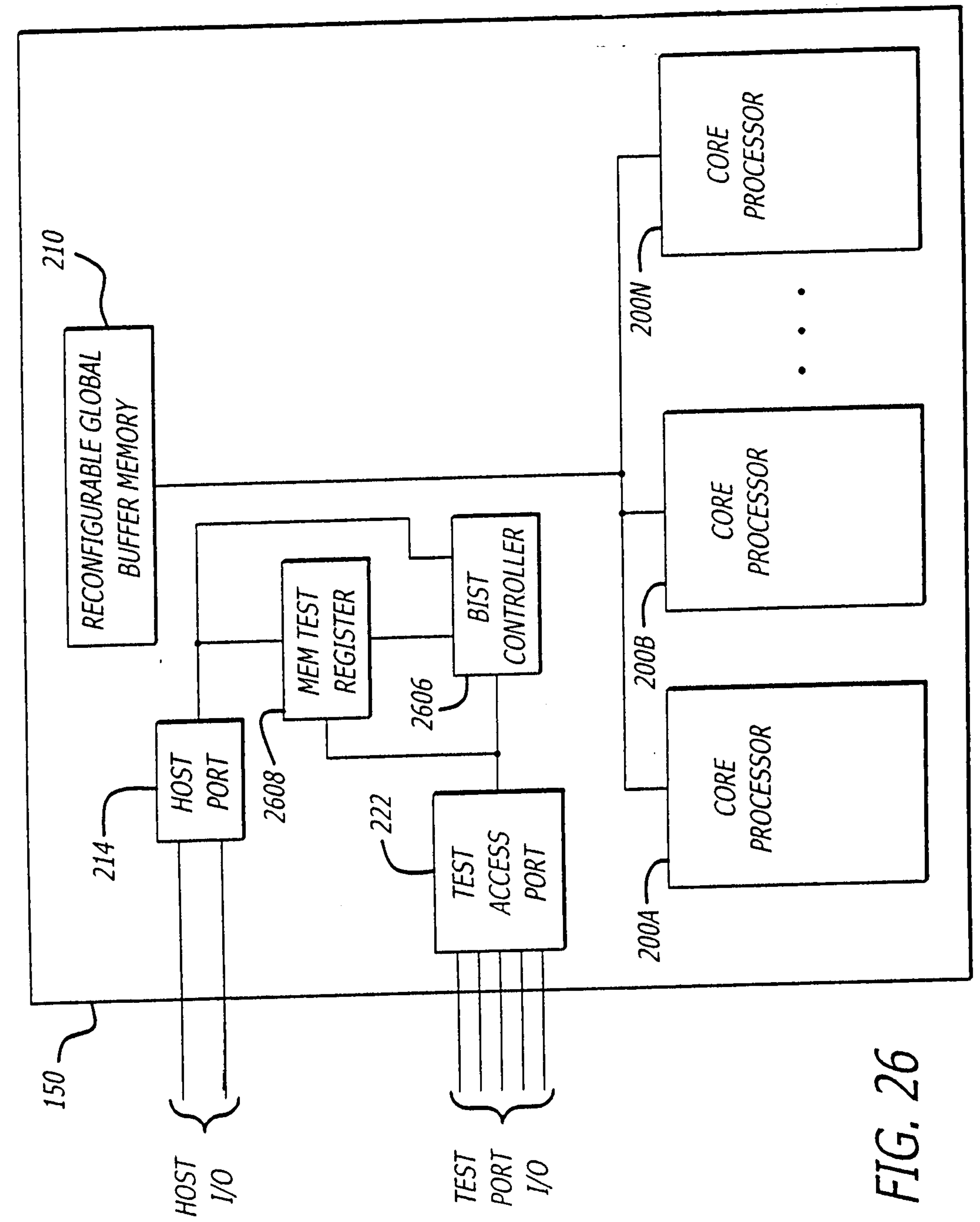


FIG. 26

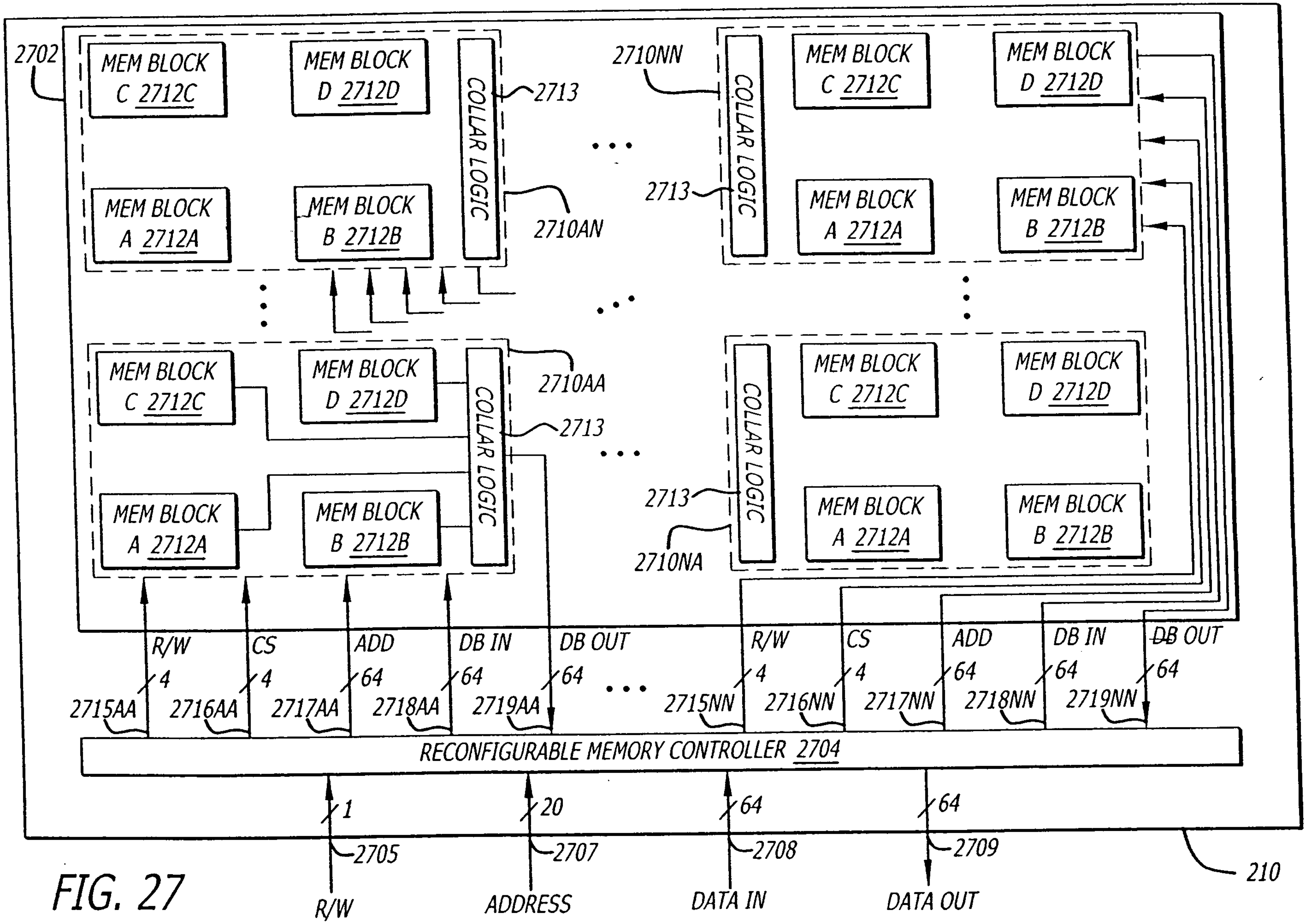

2702
MEM BLOCK C 2712C
MEM BLOCK D 2712D
COLLAR LOGIC
2713
MEM BLOCK A 2712A
MEM BLOCK B 2712B
2710AN
2710NN
2713
MEM BLOCK C 2712C
MEM BLOCK D 2712D
MEM BLOCK A 2712A
MEM BLOCK B 2712B
COLLAR LOGIC
MEM BLOCK C 2712C
MEM BLOCK D 2712D
COLLAR LOGIC
2710AA
2713
MEM BLOCK A 2712A
MEM BLOCK B 2712B
2713
2710NA
MEM BLOCK C 2712C
MEM BLOCK D 2712D
COLLAR LOGIC
MEM BLOCK A 2712A
MEM BLOCK B 2712B
R/W
4
CS
4
ADD
64
DB IN
64
DB OUT
64
2715AA
2716AA
2717AA
2718AA
2719AA
R/W
4
CS
4
ADD
64
DB IN
64
DB OUT
64
2715NN
2716NN
2717NN
2718NN
2719NN
RECONFIGURABLE MEMORY CONTROLLER 2704
1
2705
R/W
20
2707
ADDRESS
64
2708
DATA IN
64
2709
DATA OUT
210

FIG. 27

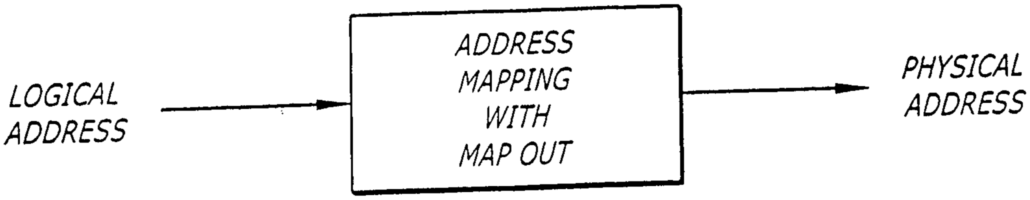


FIG. 28

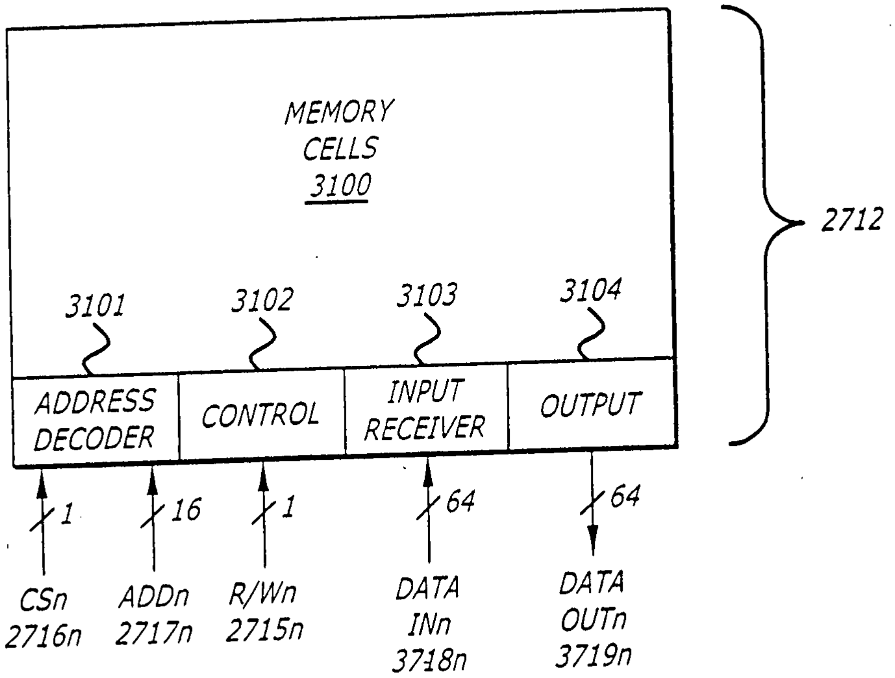


FIG. 31

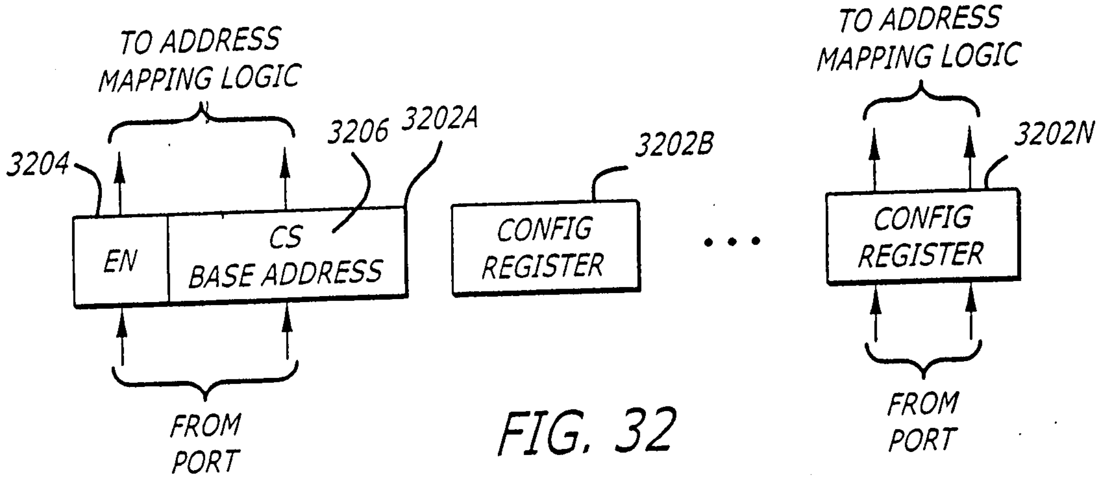
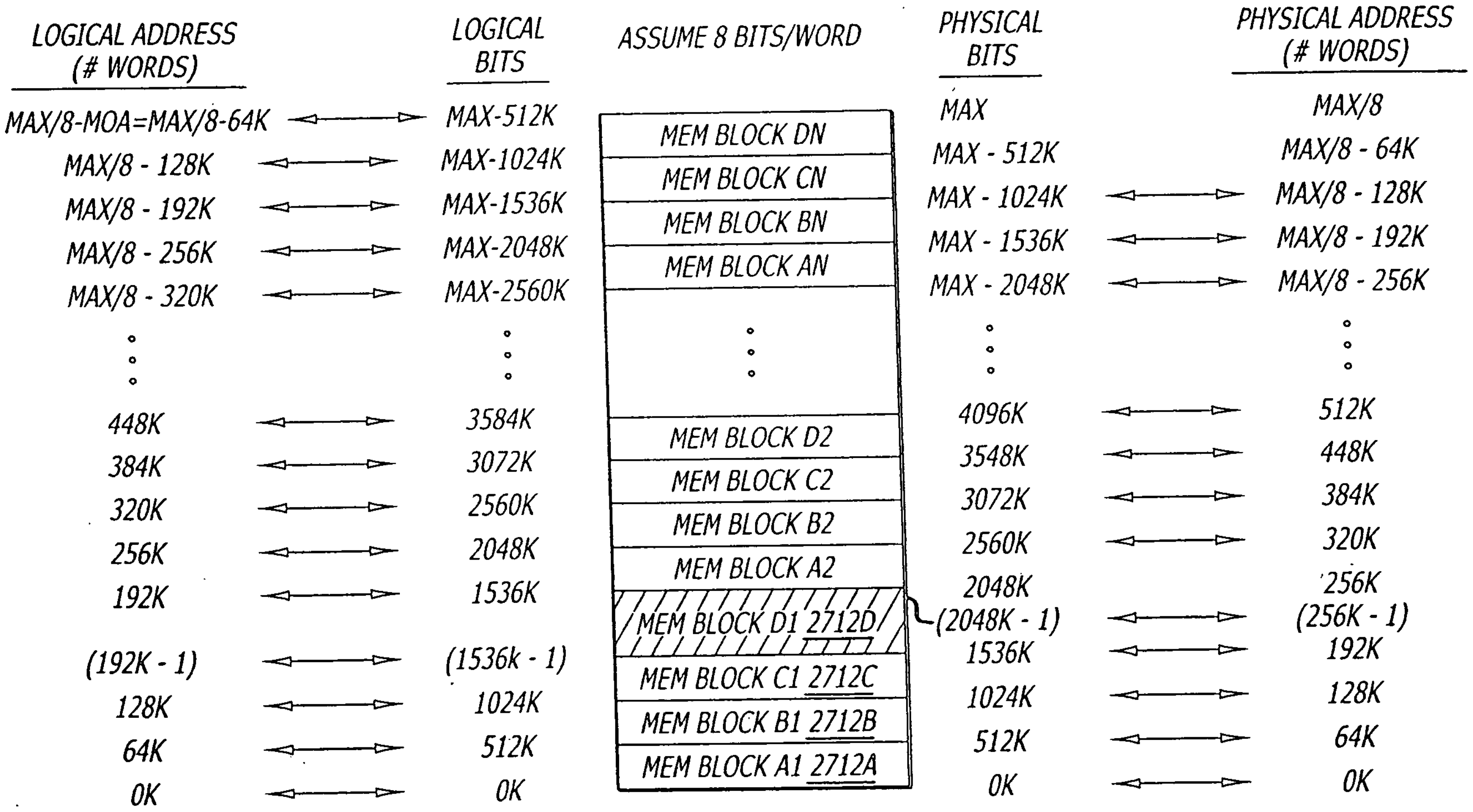


FIG. 32

FIG. 29


LOGICAL ADDRESS (# WORDS)
LOGICAL BITS
ASSUME 8 BITS/WORD
PHYSICAL BITS
PHYSICAL ADDRESS (# WORDS)
MAX/8-MOA=MAX/8-64K
MAX/8 - 128K
MAX/8 - 192K
MAX/8 - 256K
MAX/8 - 320K
448K
384K
320K
256K
192K
(192K - 1)
128K
64K
0K
MAX-512K
MAX-1024K
MAX-1536K
MAX-2048K
MAX-2560K
3584K
3072K
2560K
2048K
1536K
(1536k - 1)
1024K
512K
0K
MEM BLOCK DN
MEM BLOCK CN
MEM BLOCK BN
MEM BLOCK AN
MEM BLOCK D2
MEM BLOCK C2
MEM BLOCK B2
MEM BLOCK A2
MEM BLOCK D1 2712D
MEM BLOCK C1 2712C
MEM BLOCK B1 2712B
MEM BLOCK A1 2712A
MAX
MAX - 512K
MAX - 1024K
MAX - 1536K
MAX - 2048K
4096K
3548K
3072K
2560K
2048K
(2048K - 1)
1536K
1024K
512K
0K
MAX/8
MAX/8 - 64K
MAX/8 - 128K
MAX/8 - 192K
MAX/8 - 256K
512K
448K
384K
320K
256K
(256K - 1)
192K
128K
64K
0K

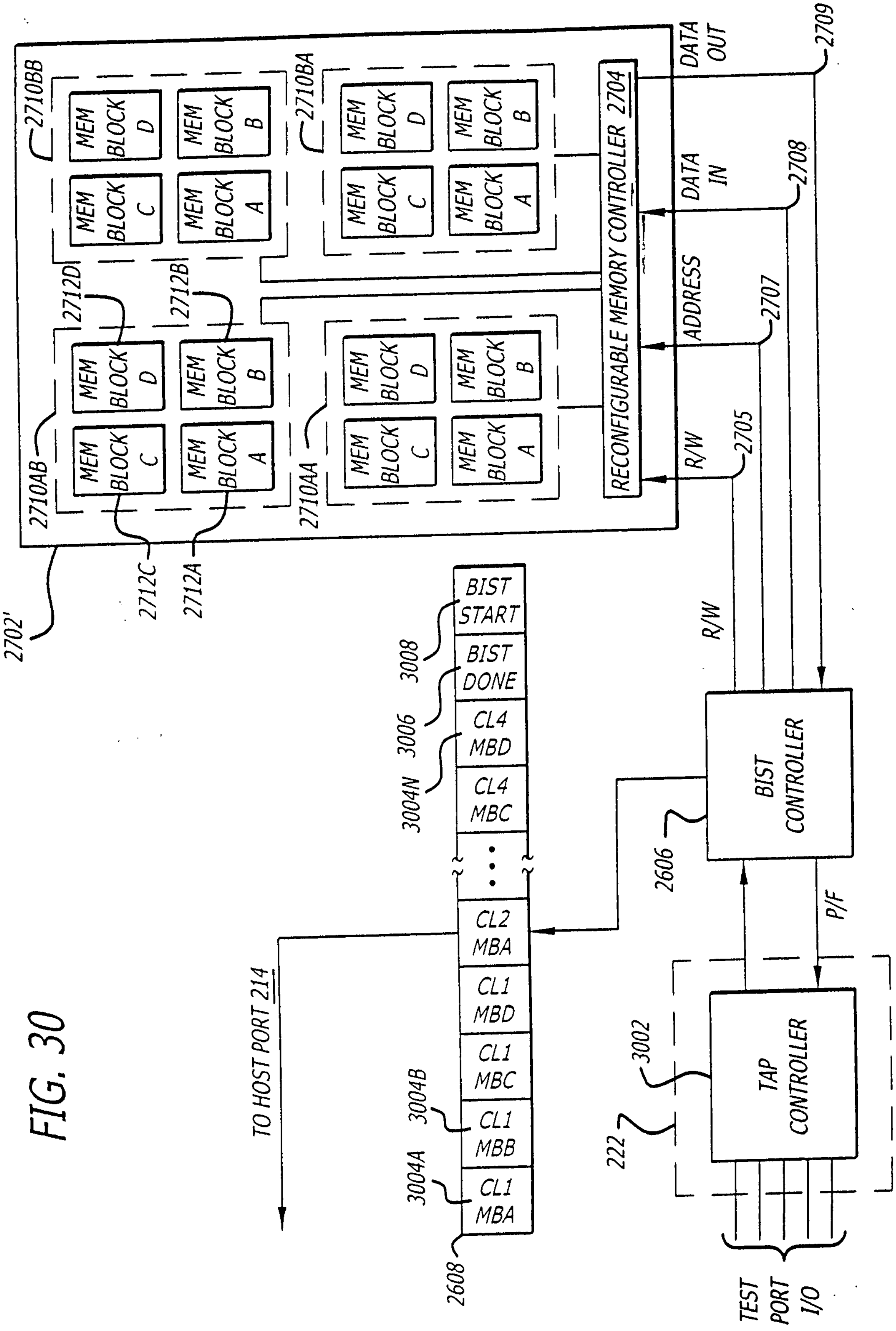

2710BB
MEM BLOCK D
MEM BLOCK B
MEM BLOCK C
MEM BLOCK A
2710BA
MEM BLOCK D
MEM BLOCK B
MEM BLOCK C
MEM BLOCK A
2712D
2712B
2710AB
MEM BLOCK D
MEM BLOCK B
MEM BLOCK C
MEM BLOCK A
2710AA
MEM BLOCK D
MEM BLOCK B
MEM BLOCK C
MEM BLOCK A
RECONFIGURABLE MEMORY CONTROLLER 2704
DATA OUT
2709
DATA IN
2708
ADDRESS
2707
R/W
2705
2702'
2712C
2712A
BIST START
BIST DONE
CL4 MBD
CL4 MBC
CL2 MBA
CL1 MBD
CL1 MBC
CL1 MBB
CL1 MBA
3008
3006
3004N
3004B
3004A
2608
TO HOST PORT 214
R/W
BIST CONTROLLER
2606
P/F
TAP CONTROLLER
3002
222
TEST PORT I/O

FIG. 30

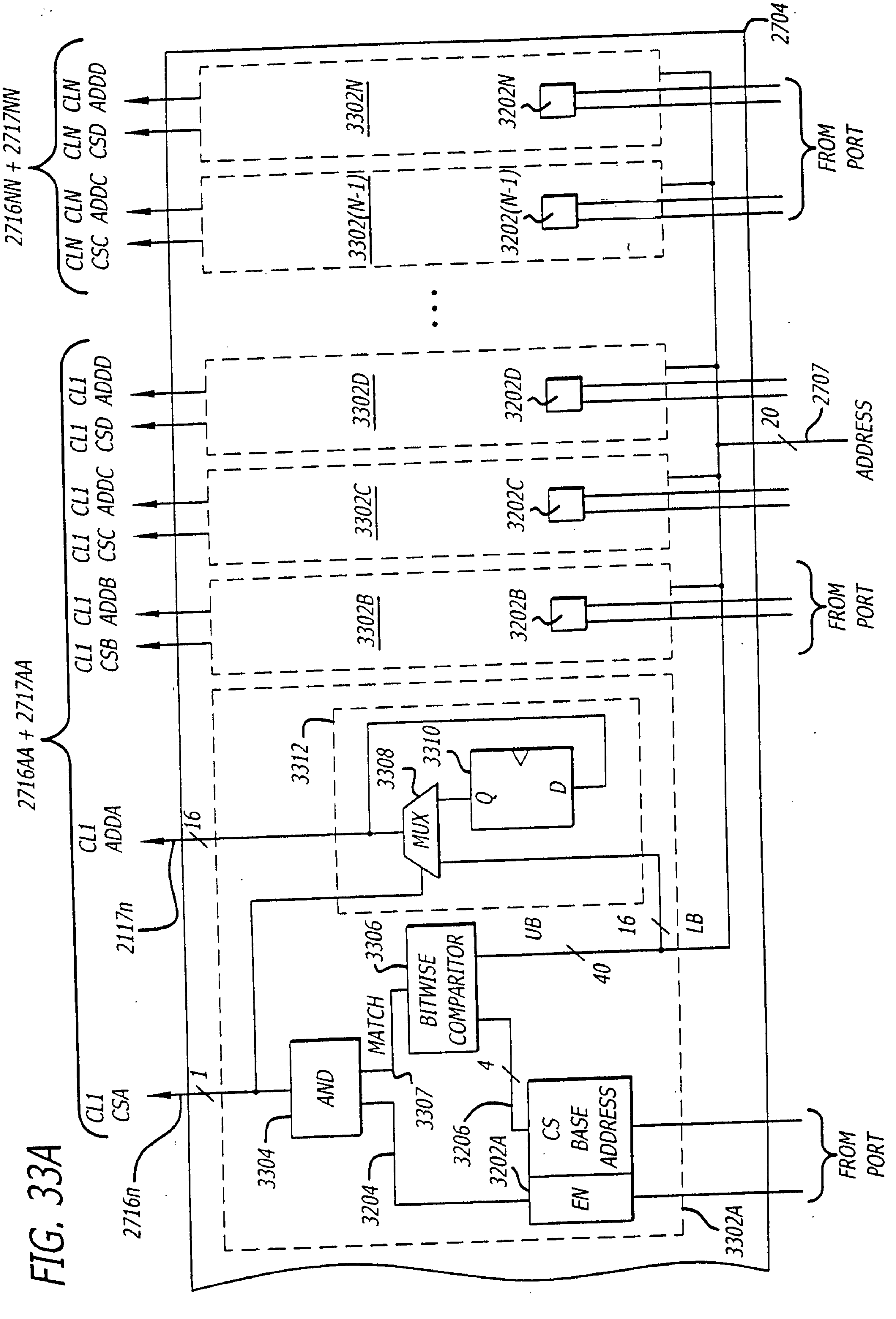

2716NN + 2717NN
CLN CSC
CLN ADDC
CLN CSD
CLN ADDD
2704
3302N
3202N
3302(N-1)
3202(N-1)
FROM PORT
2716AA + 2717AA
CL1 CSB
CL1 ADDB
CL1 CSC
CL1 ADDC
CL1 CSD
CL1 ADDD
3302D
3202D
3302C
3202C
3302B
3202B
20
2707
ADDRESS
FROM PORT
CL1 ADDA
16
2117n
3312
3308
3310
MUX
Q
D
3306
BITWISE COMPARITOR
MATCH
UB
LB
16
40
4
CL1 CSA
1
AND
3304
3307
3206
3202A
CS BASE ADDRESS
EN
3204
2716n
3302A
FROM PORT

FIG. 33A

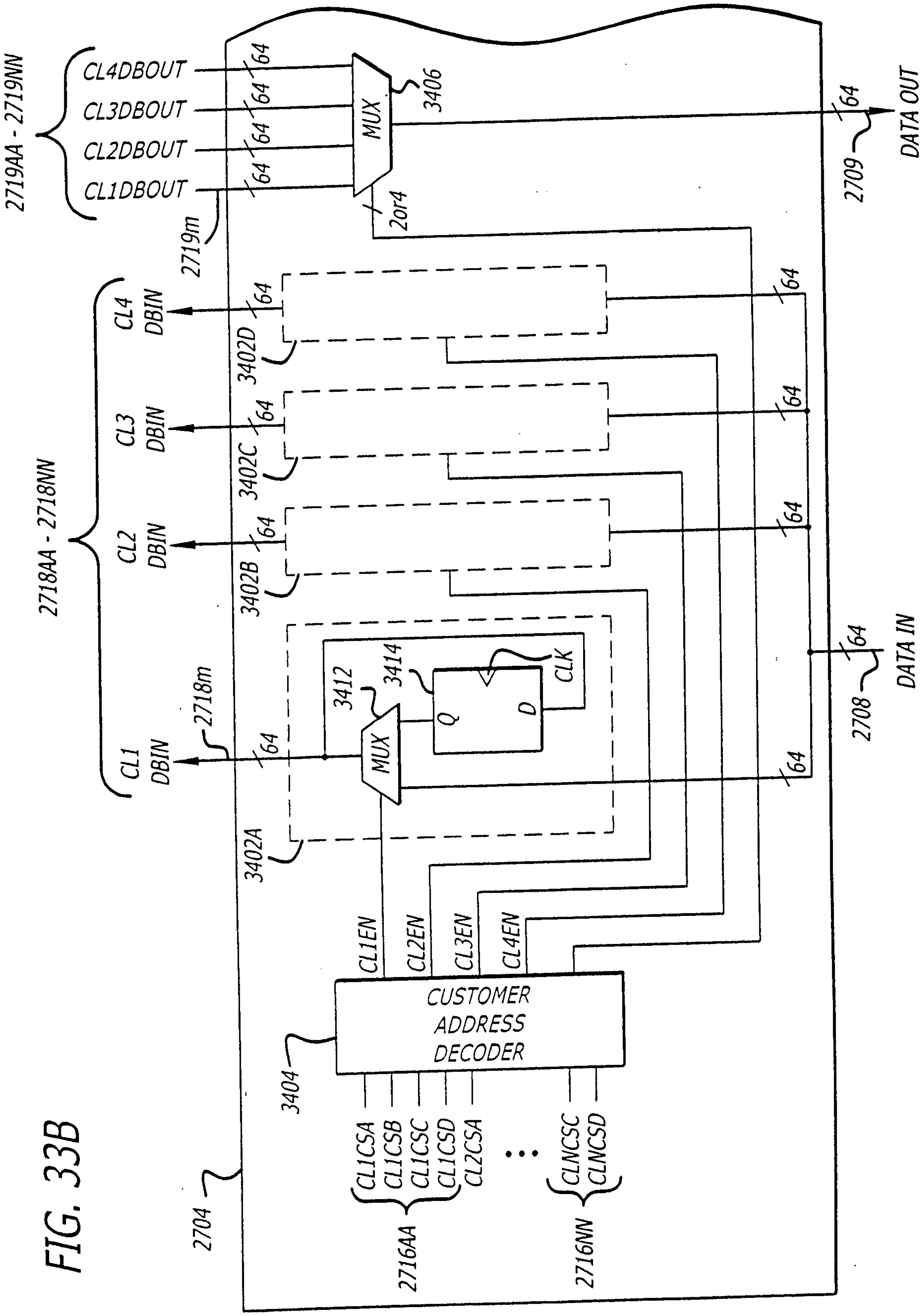

2719AA - 2719NN
CL4DBOUT
CL3DBOUT
CL2DBOUT
CL1DBOUT
2719m
64
64
64
64
MUX
3406
2or4
64
DATA OUT
2709
2718AA - 2718NN
CL4 DBIN
CL3 DBIN
CL2 DBIN
CL1 DBIN
2718m
3402D
3402C
3402B
3402A
64
64
64
64
64
64
64
64
64
DATA IN
2708
3412
3414
MUX
Q
D
CLK
CL1EN
CL2EN
CL3EN
CL4EN
CUSTOMER ADDRESS DECODER
3404
CL1CSA
CL1CSB
CL1CSC
CL1CSD
CL2CSA
...
CLNCSC
CLNCSD
2716AA
2716NN
2704

FIG. 33B

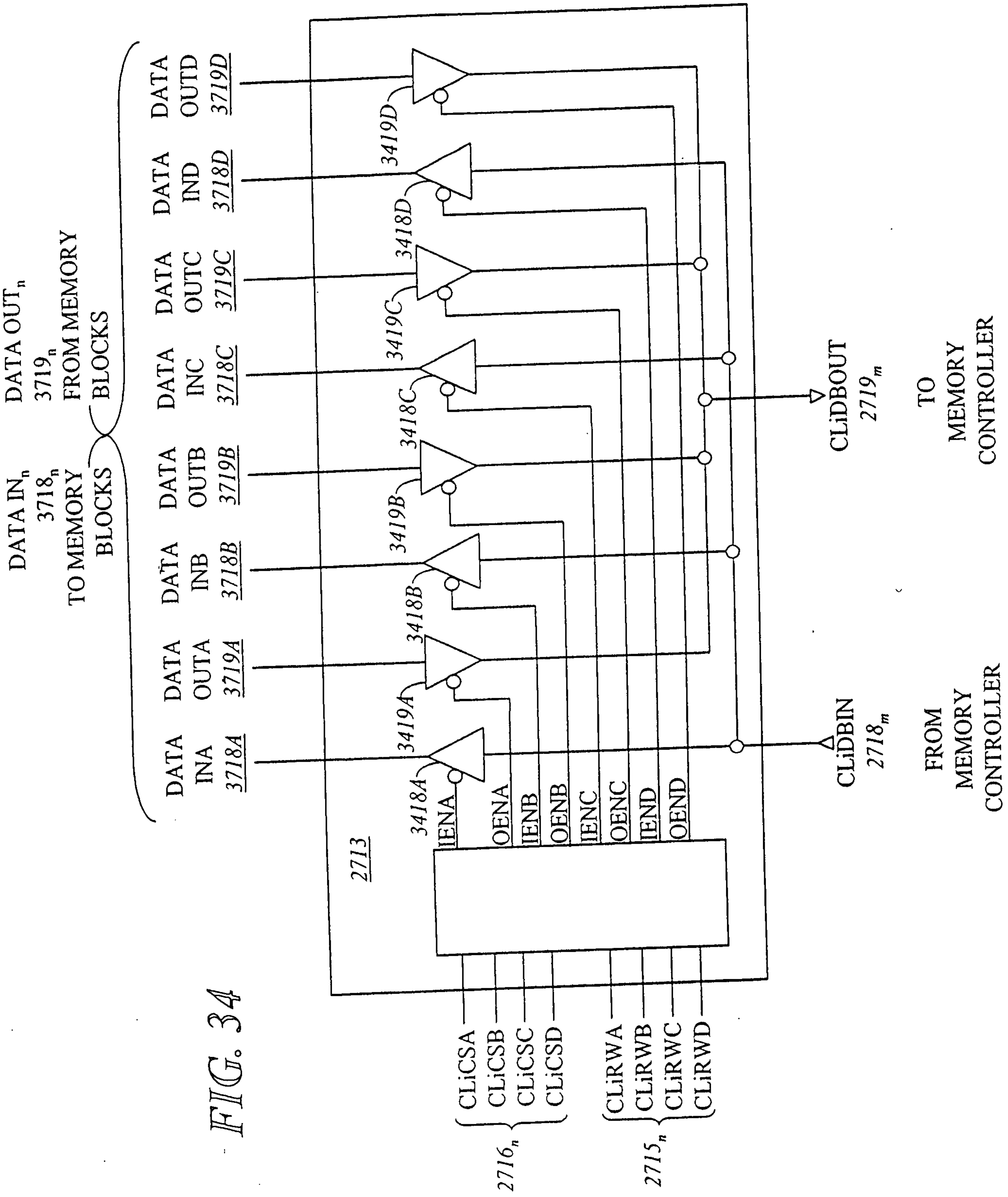

DATA IN$_n$
3718$_n$
TO MEMORY
BLOCKS
DATA OUT$_n$
3719$_n$
FROM MEMORY
BLOCKS
DATA INA 3718A
DATA OUTA 3719A
DATA INB 3718B
DATA OUTB 3719B
DATA INC 3718C
DATA OUTC 3719C
DATA IND 3718D
DATA OUTD 3719D
3418A
3419A
3418B
3419B
3418C
3419C
3418D
3419D
IENA
OENA
IENB
OENB
IENC
OENC
IEND
OEND
2713
CLiDBIN
2718$_m$
FROM MEMORY CONTROLLER
CLiDBOUT
2719$_m$
TO MEMORY CONTROLLER
CLiCSA
CLiCSB
CLiCSC
CLiCSD
2716$_n$
CLiRWA
CLiRWB
CLiRWC
CLiRWD
2715$_n$

FIG. 34

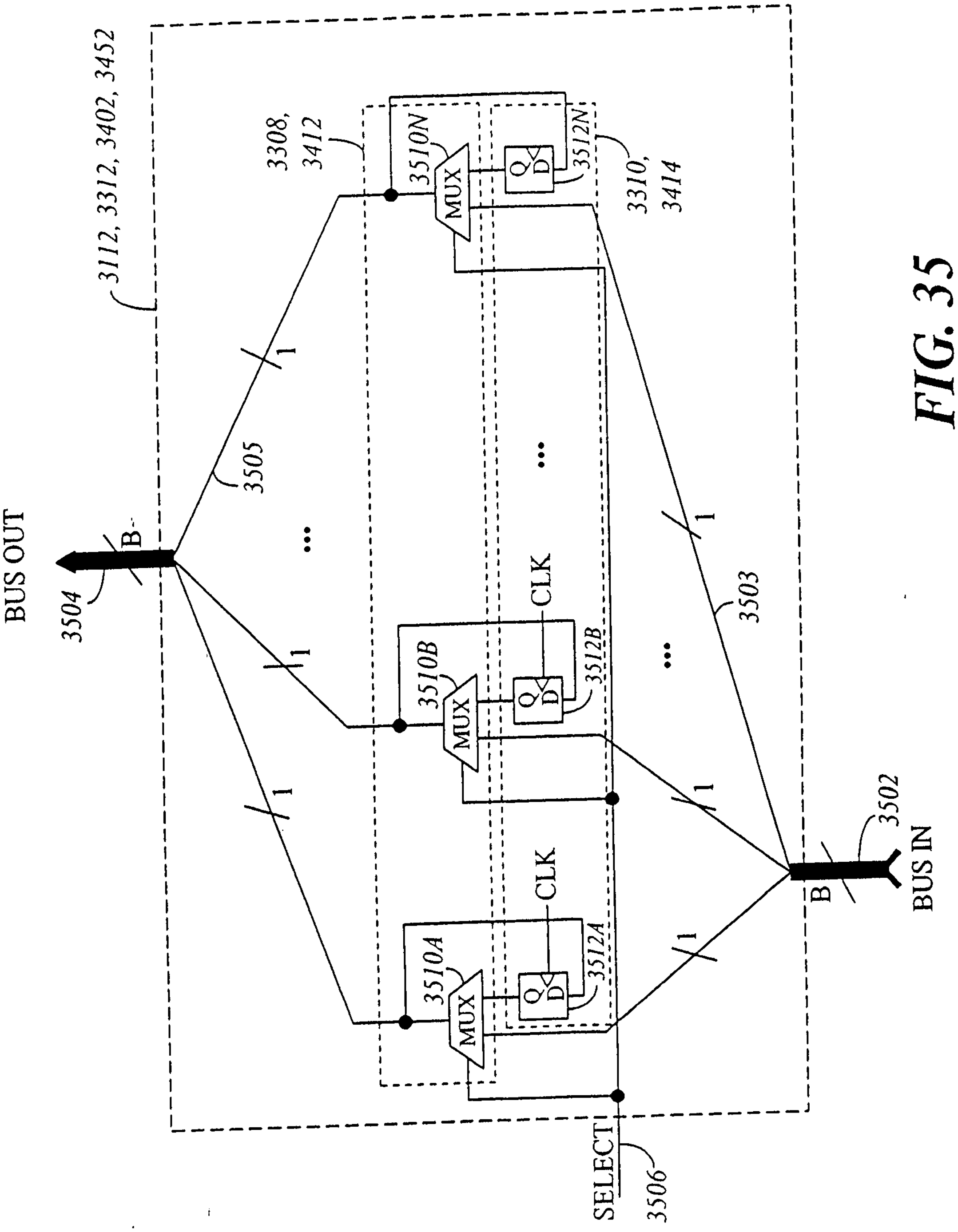

3112, 3312, 3402, 3452
3308, 3412
3310, 3414
BUS OUT
3504
B
3505
1
1
1
3510N
MUX
Q
D
3512N
3510B
MUX
Q
D
CLK
3512B
3510A
MUX
Q
D
CLK
3512A
1
1
1
3503
B
3502
BUS IN
SELECT
3506

FIG. 35

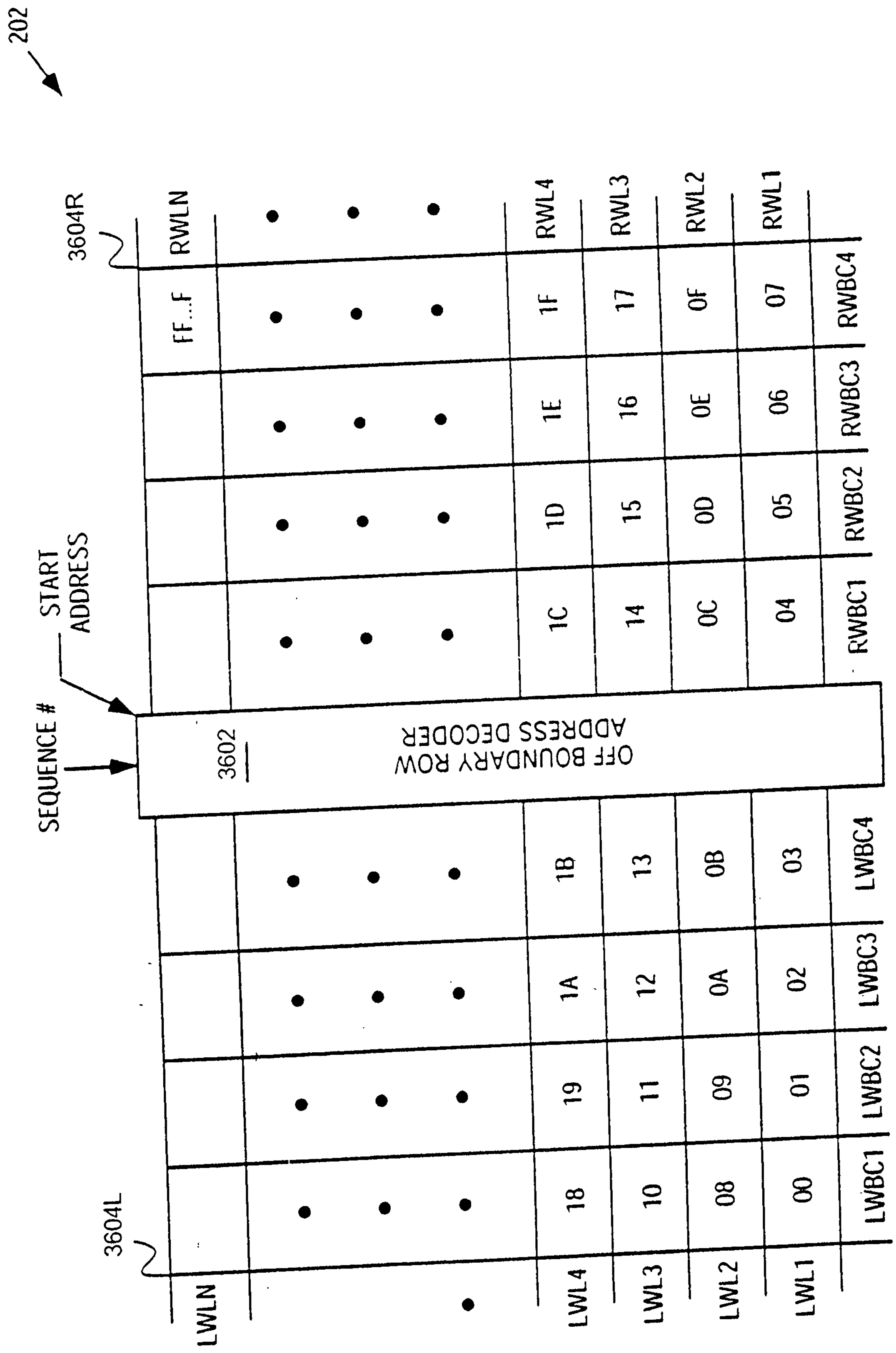

202
3604R
3604L
START ADDRESS
SEQUENCE #
3602
OFF BOUNDARY ROW ADDRESS DECODER
RWLN
RWL4
RWL3
RWL2
RWL1
LWLN
LWL4
LWL3
LWL2
LWL1
RWBC1
RWBC2
RWBC3
RWBC4
LWBC1
LWBC2
LWBC3
LWBC4
FF...F
1C
1D
1E
1F
14
15
16
17
0C
0D
0E
0F
04
05
06
07
18
19
1A
1B
10
11
12
13
08
09
0A
0B
00
01
02
03

FIG. 36A

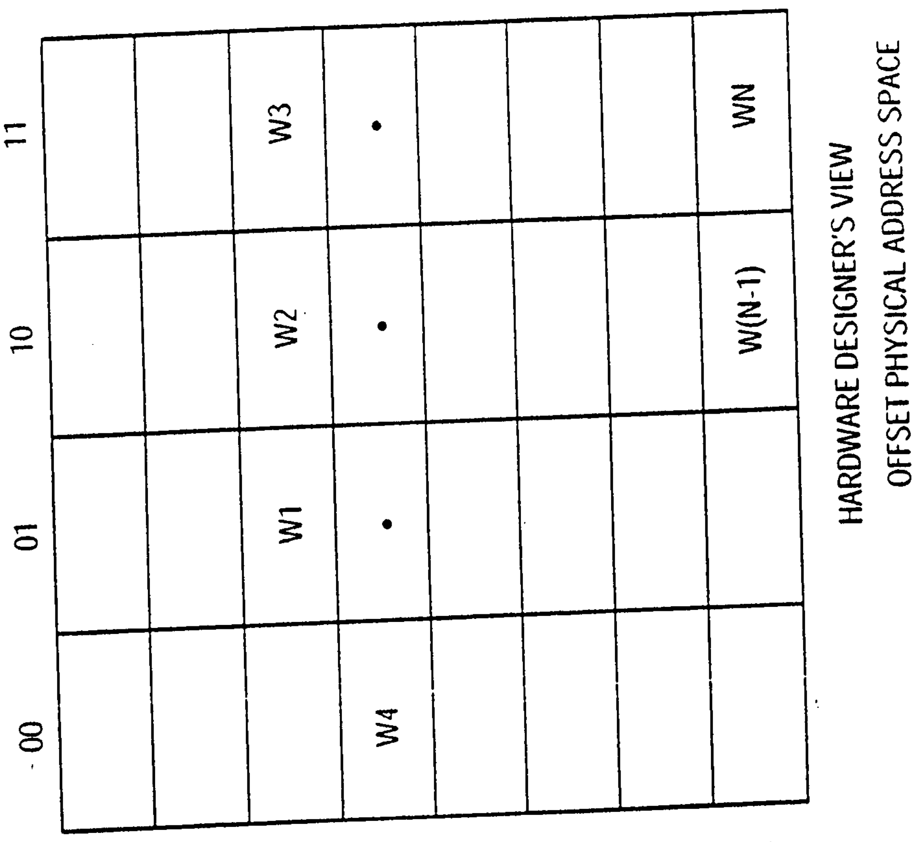


FIG. 36C

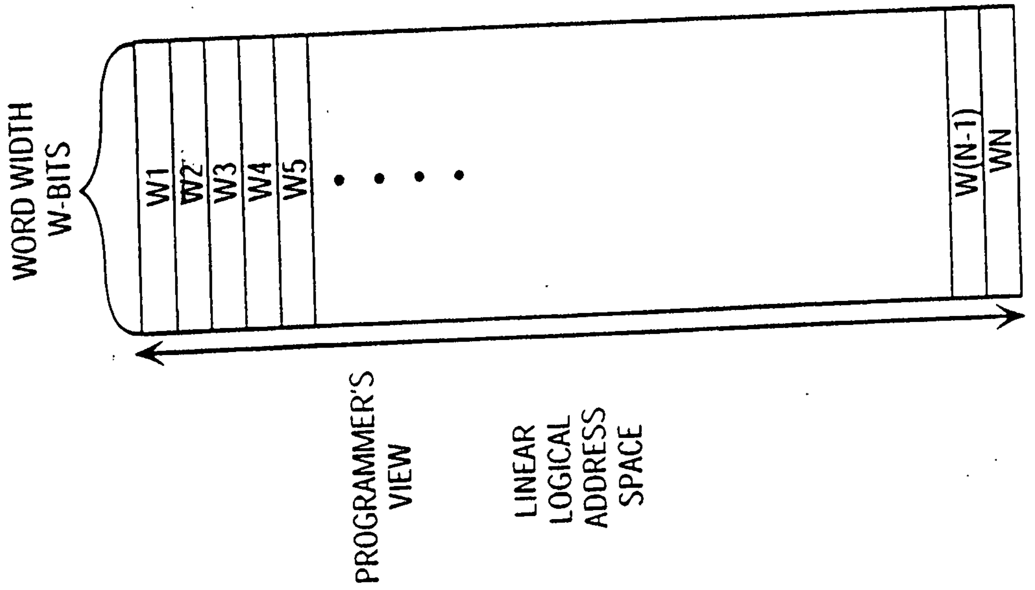


FIG. 36B

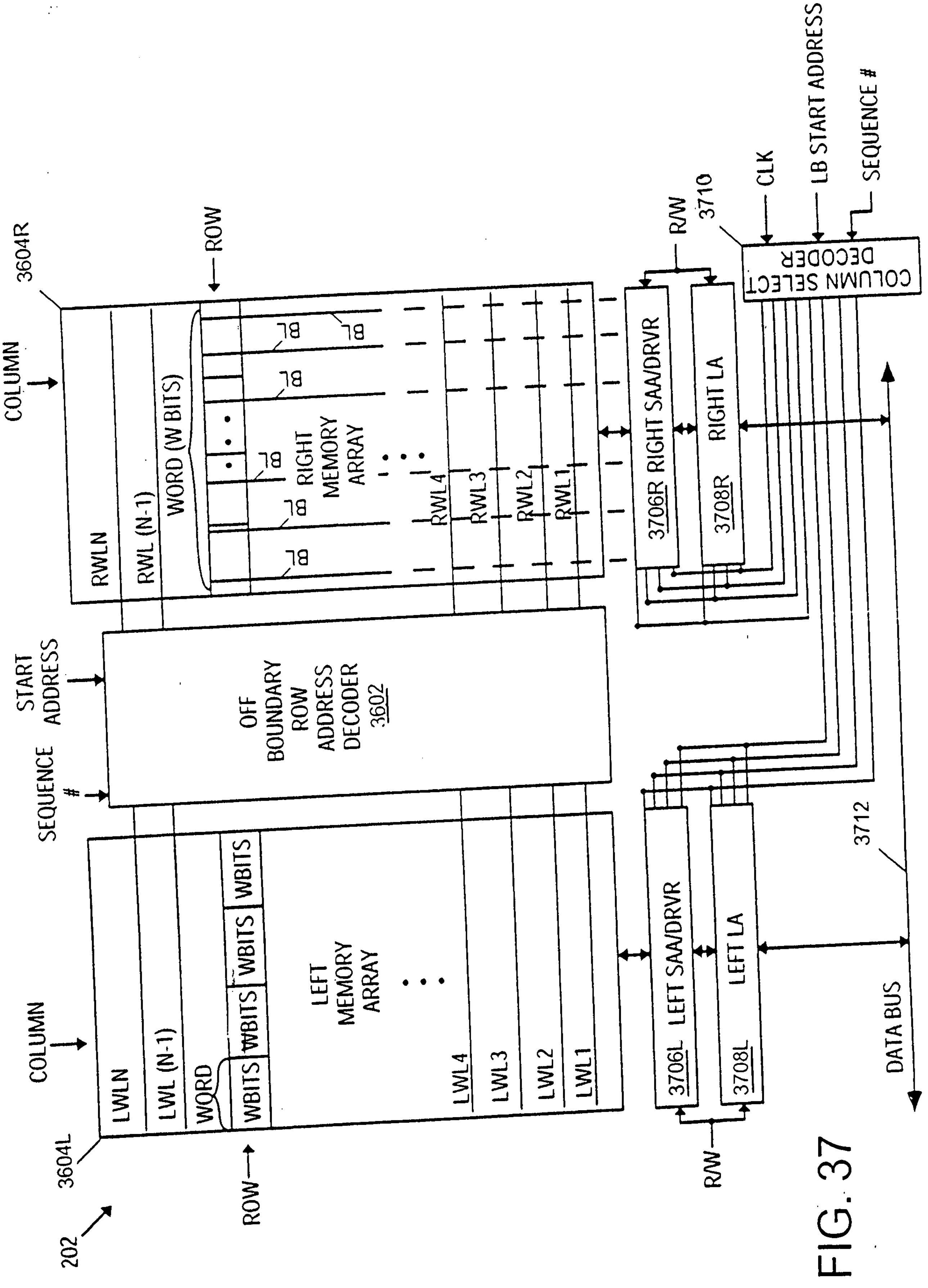

202
3604L
ROW
COLUMN
LWLN
LWL (N-1)
WORD
WBITS
WBITS
WBITS
WBITS
WBITS
LEFT MEMORY ARRAY
LWL4
LWL3
LWL2
LWL1
SEQUENCE #
START ADDRESS
OFF BOUNDARY ROW ADDRESS DECODER 3602
3604R
COLUMN
ROW
RWLN
RWL (N-1)
WORD (W BITS)
BL
BL
BL
BL
BL
BL
RIGHT MEMORY ARRAY
RWL4
RWL3
RWL2
RWL1
R/W
3706L LEFT SAA/DRVR
3708L LEFT LA
3706R RIGHT SAA/DRVR
3708R RIGHT LA
R/W
3710
CLK
LB START ADDRESS
SEQUENCE #
COLUMN SELECT DECODER
3712
DATA BUS

FIG. 37

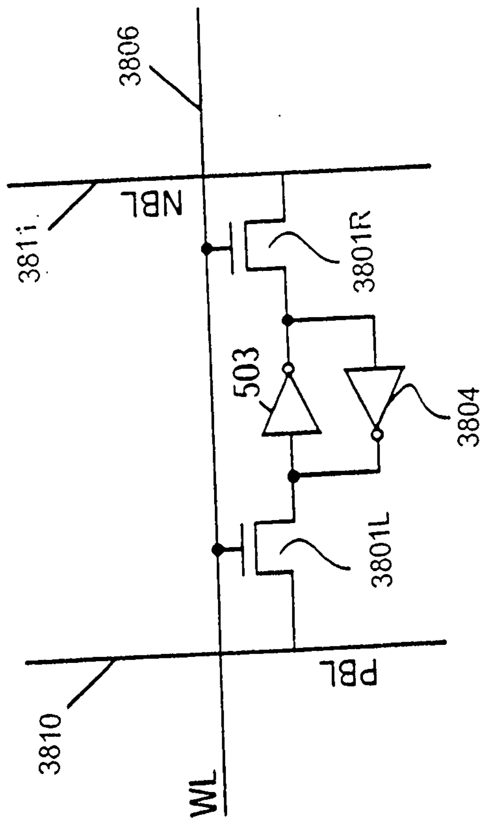


FIG. 38A

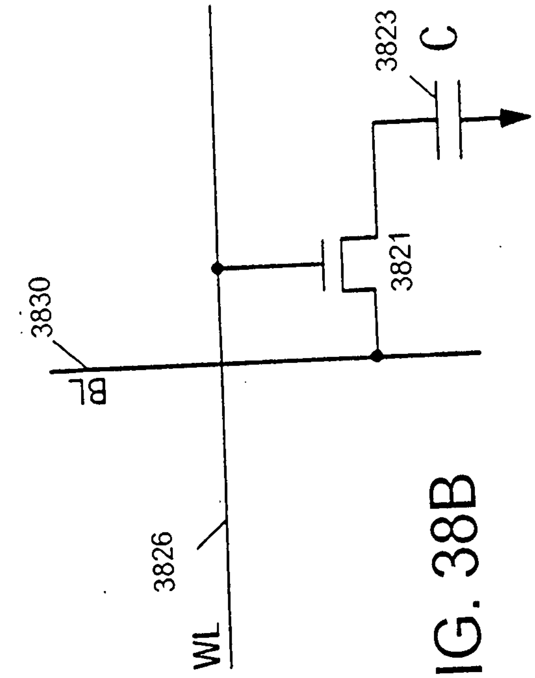


FIG. 38B

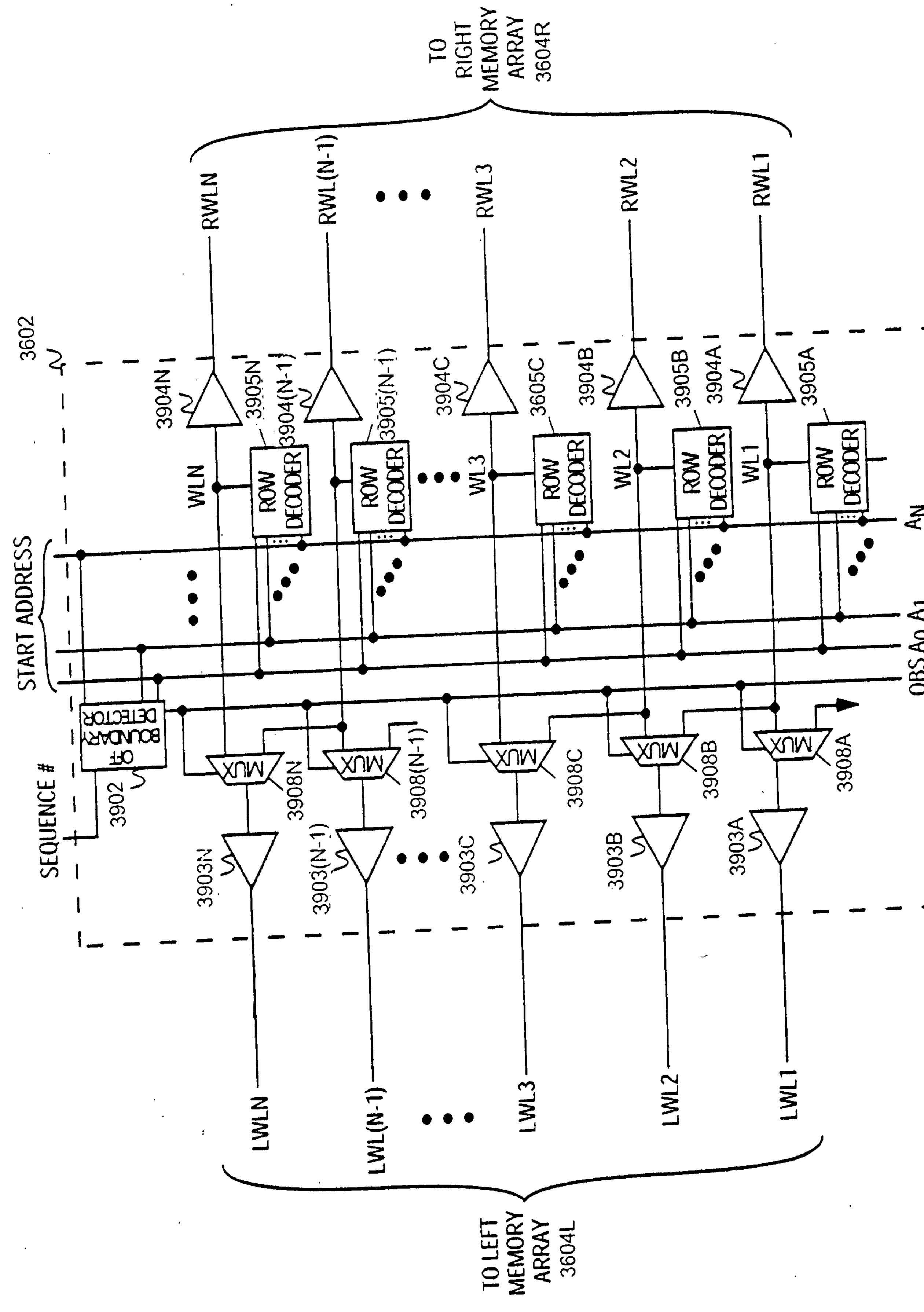
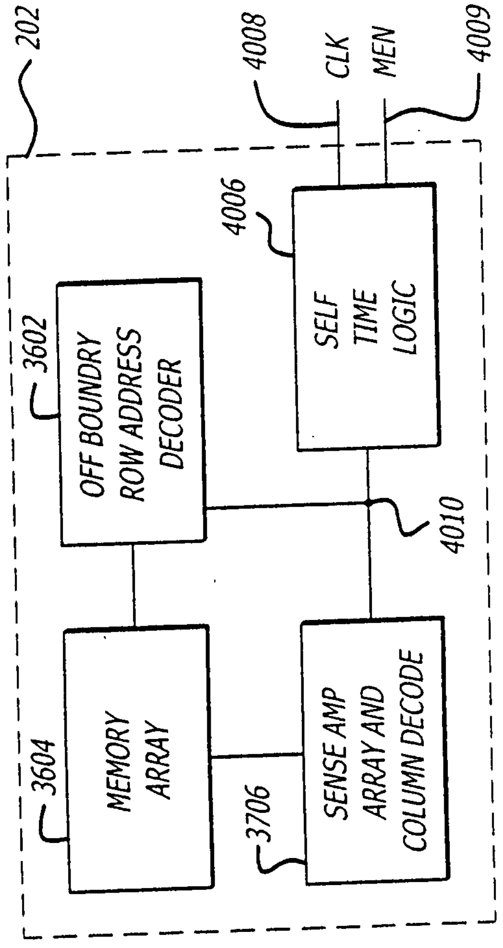

TO RIGHT MEMORY ARRAY 3604R
RWLN
RWL(N-1)
RWL3
RWL2
RWL1
3602
3904N
3905N
3904(N-1)
3905(N-1)
3904C
3605C
3904B
3905B
3904A
3905A
WLN
WL3
WL2
WL1
ROW DECODER
START ADDRESS
AN
A1
A0
OBS
3910
SEQUENCE #
OFF BOUNDARY DETECTOR
3902
MUX
3908N
3908(N-1)
3908C
3908B
3908A
3903N
3903(N-1)
3903C
3903B
3903A
LWLN
LWL(N-1)
LWL3
LWL2
LWL1
TO LEFT MEMORY ARRAY 3604L

FIG. 39

Blakely, Sokoloff, Taylor & Zafman LLP (714) 557-3800
Title: Methods for Power Reduction in a Digital Signal Processor Integrated Circuit
1st Named Inventor: Ruban Kanapathippillai
Application No.: 10/645,366 Docket No.: 42P14037D
Sheet: 60 of 64



202
4008
CLK
MEN
4009
4006
SELF TIME LOGIC
3602
OFF BOUNDRY ROW ADDRESS DECODER
4010
3604
MEMORY ARRAY
3706
SENSE AMP ARRAY AND COLUMN DECODE

FIG. 40

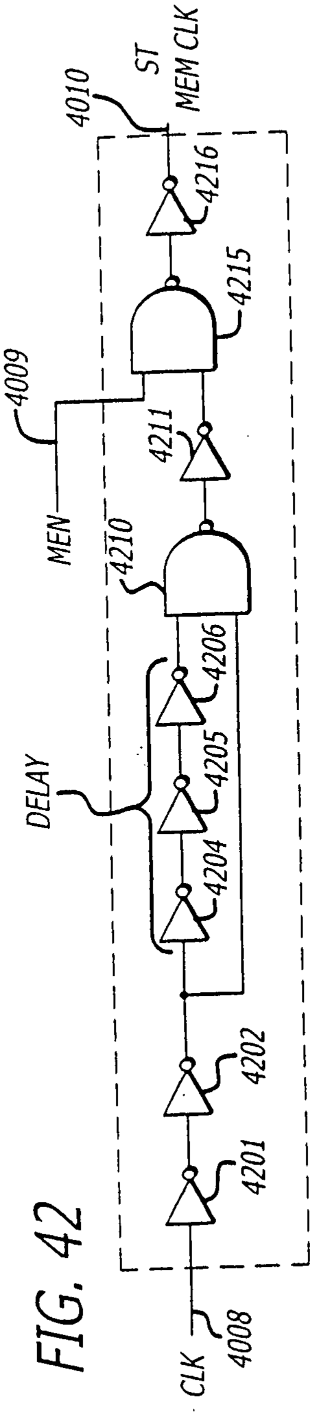


FIG. 42

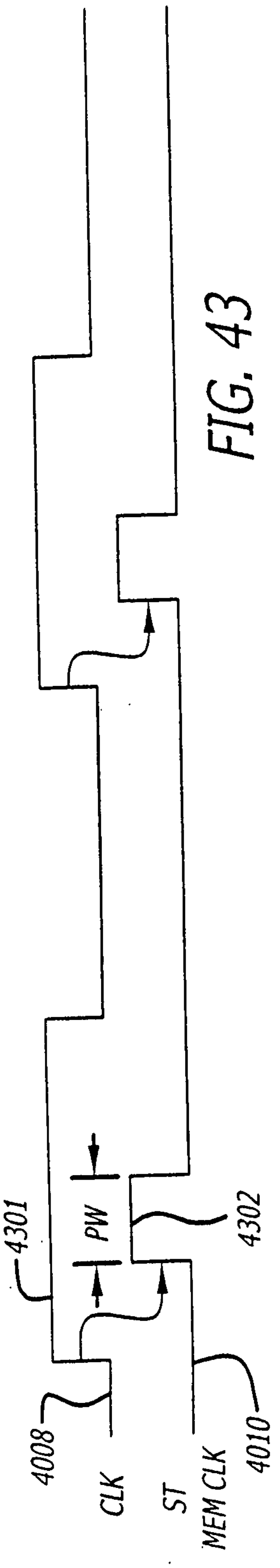
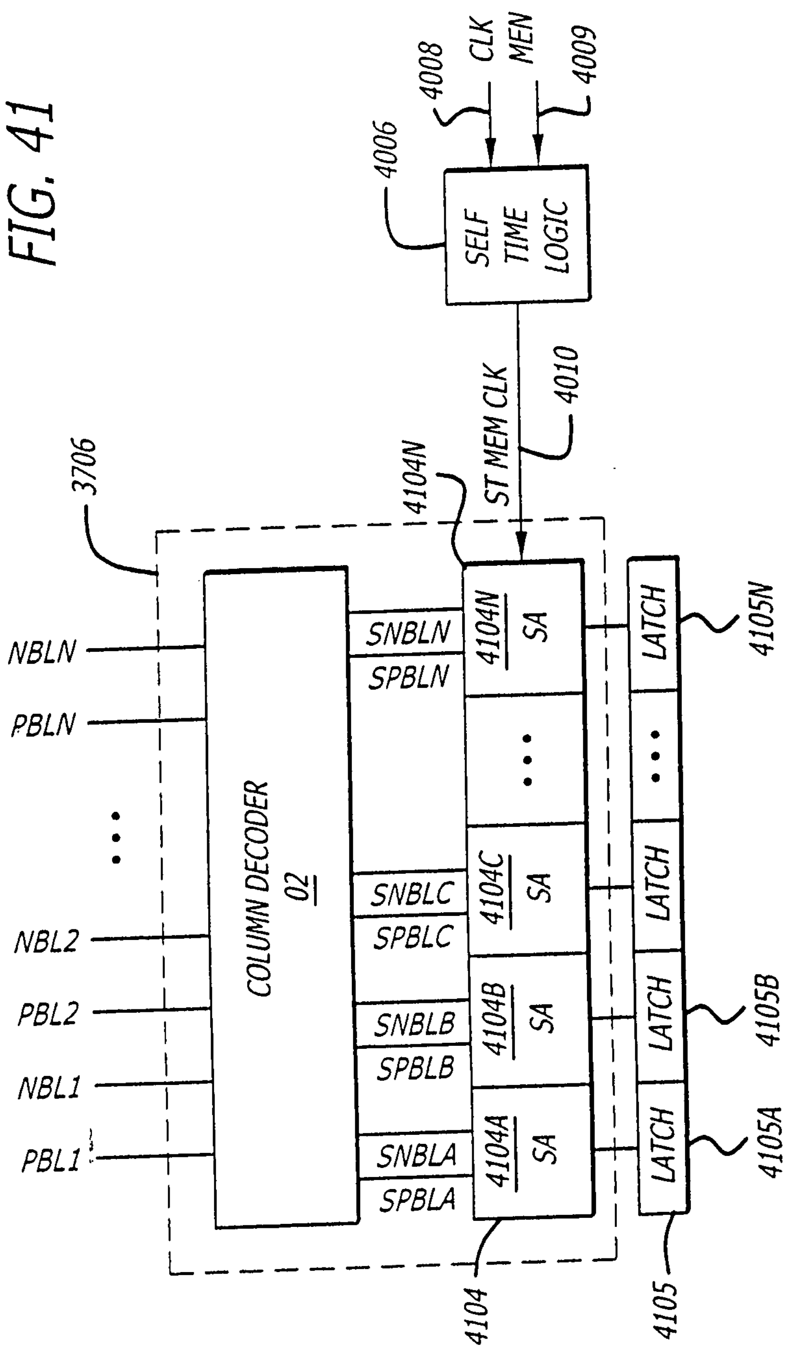


FIG. 43

FIG. 41


4008
CLK
MEN
4009
4006
SELF TIME LOGIC
ST MEM CLK
4010
3706
4104N
NBLN
PBLN
NBL2
PBL2
NBL1
PBL1
COLUMN DECODER 02
SNBLN
SPBLN
SNBLC
SPBLC
SNBLB
SPBLB
SNBLA
SPBLA
4104N SA
4104C SA
4104B SA
4104A SA
LATCH
LATCH
LATCH
LATCH
4105N
4105B
4105A
4104
4105

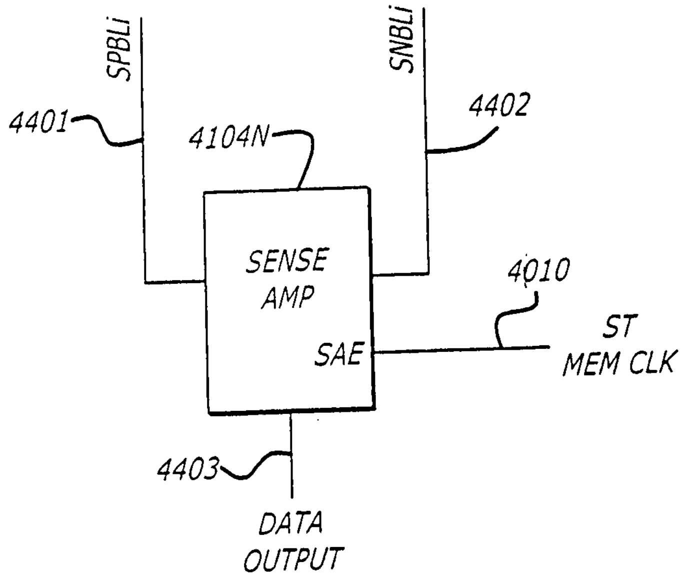


FIG. 44A

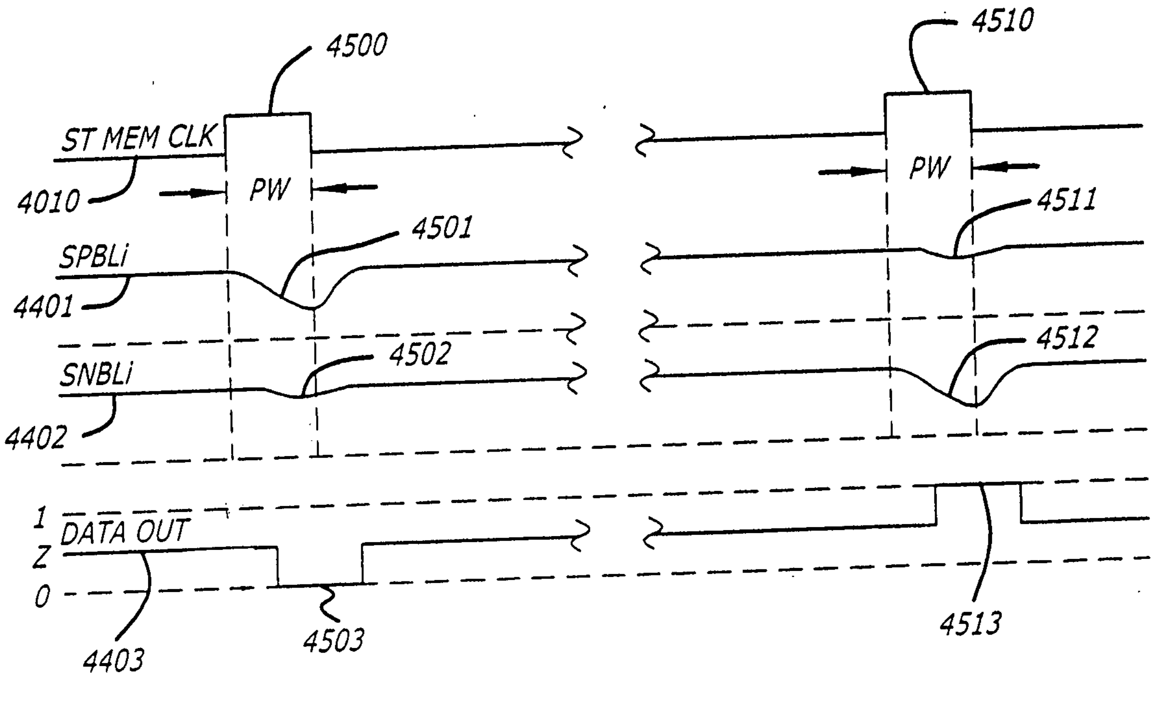


FIG. 45

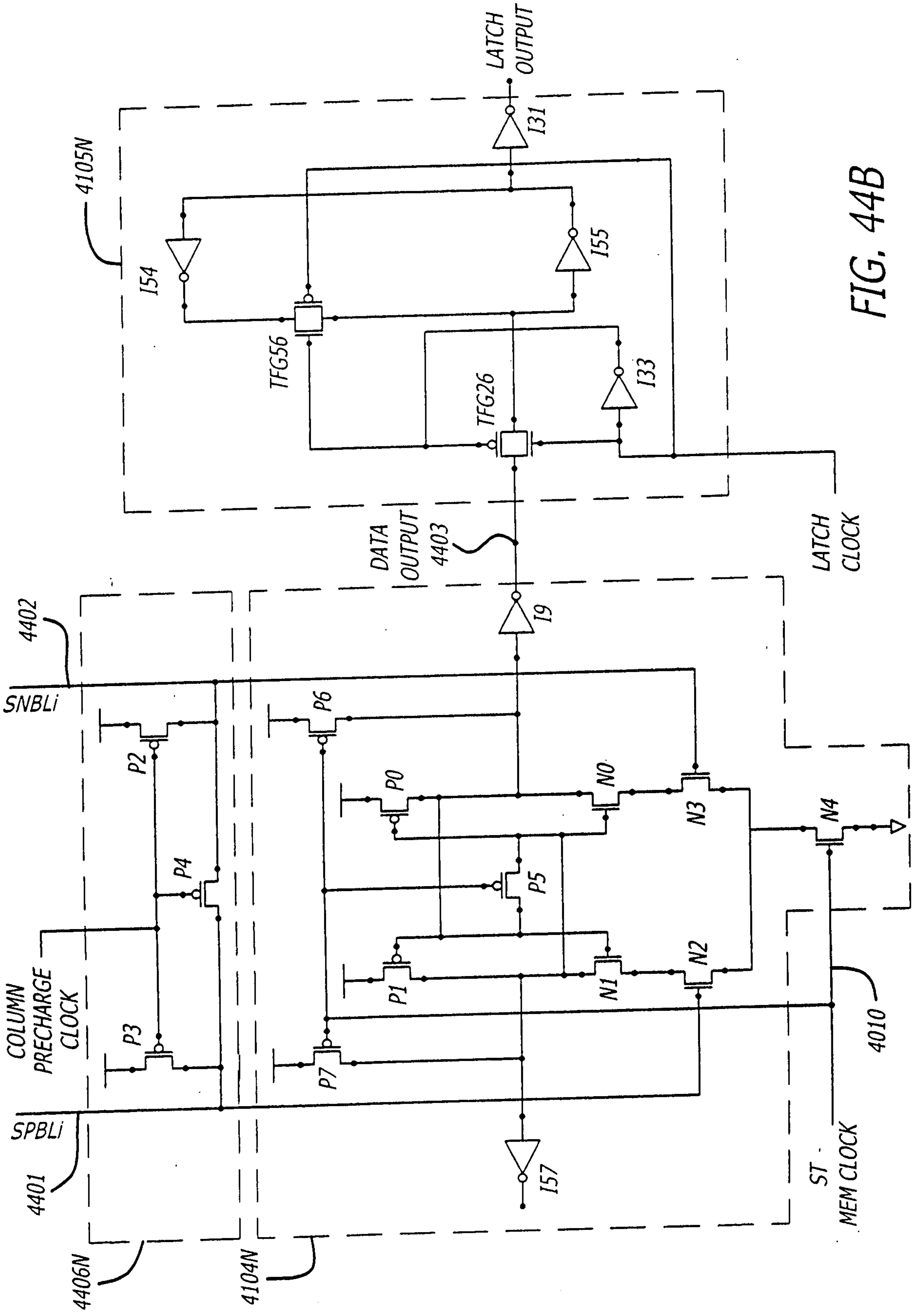

LATCH OUTPUT
I31
4105N
I54
I55
TFG56
TFG26
I33
DATA OUTPUT 4403
LATCH CLOCK
4402
SNBLi
I9
P2
P6
P0
N0
N3
P4
P5
N4
COLUMN PRECHARGE CLOCK
P1
N1
N2
4010
P3
P7
SPBLi
4401
I57
ST MEM CLOCK
4406N
4104N

FIG. 44B

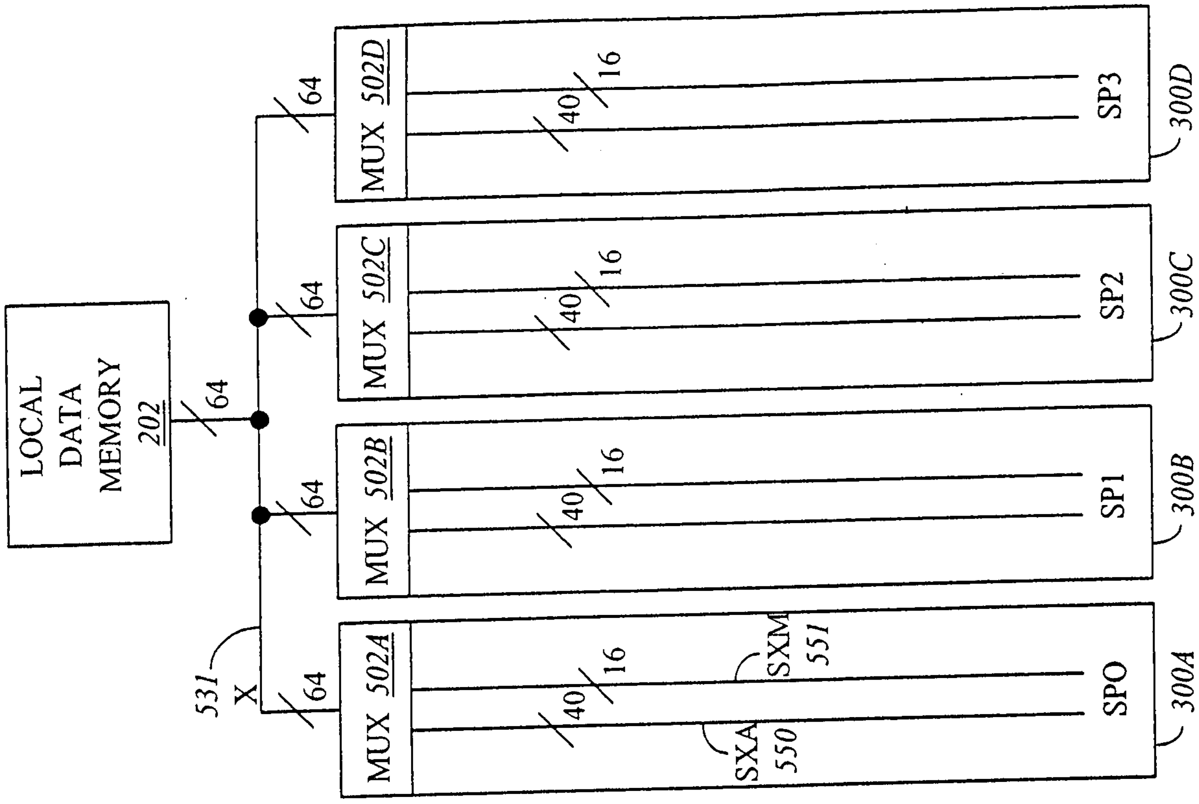


FIG. 46B

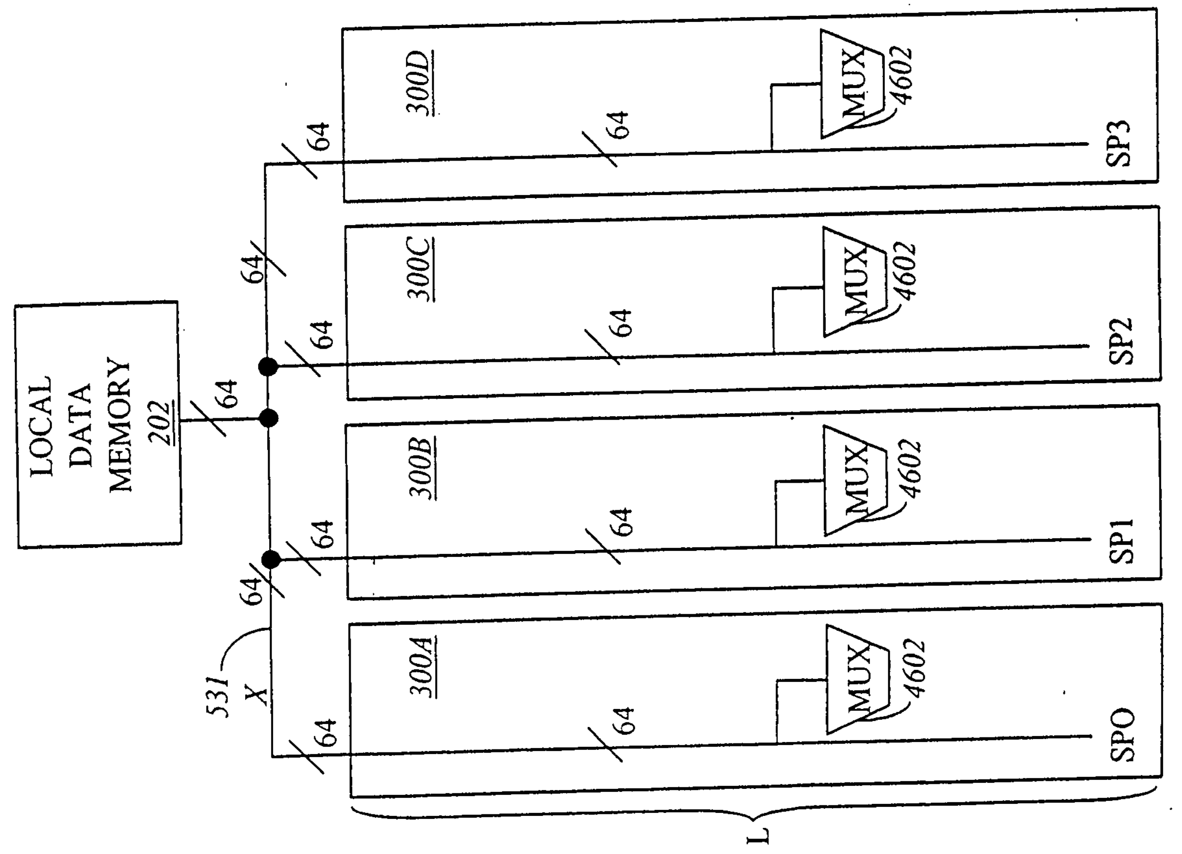


FIG. 46A